

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas Carrera de Software

Desarrollo de una aplicación móvil para reconocimiento facial con realidad aumentada utilizando Unity y AR Foundation que permita la identificación de personas dentro del campus de la Universidad Técnica del Norte.

Trabajo de grado previo a la obtención del título de Ingeniero de Software presentado ante la ilustre Universidad Técnica del Norte.

Autor:

Roberth Gabriel Lima Carvajal

Director:

Ing. Fausto Alberto Salazar Fierro, MSc

Ibarra – Ecuador

2025



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	2100572755		
APELLIDOS Y NOMBRES:	Lima Carvajal Roberth Gabriel		
DIRECCIÓN:	Ibarra, El Olivo		
EMAIL:	rglimac@utn.edu.ec		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0967732818

DATOS DE LA OBRA	
TÍTULO:	Desarrollo de una aplicación móvil para reconocimiento facial con realidad aumentada utilizando Unity y AR Foundation que permita la identificación de personas dentro del campus de la Universidad Técnica del Norte.
AUTOR(ES):	Lima Carvajal Roberth Gabriel
FECHA:	17 de feb. de 2025
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSTGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO DE SOFTWARE
DIRECTOR:	ING. Fausto Alberto Salazar Fierro, MSC.
ASESOR 1:	ING. Carpio Agapito Pineda Manosalvas, MSC.

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 17 días del mes de febrero de 2025

EL AUTOR:



ESTUDIANTE

Roberth Gabriel Lima Carvajal

C.I 2100572755

CERTIFICACIÓN DIRECTOR

Ibarra 17 de febrero del 2025

CERTIFICACIÓN DIRECTOR DEL TRABAJO DE TITULACIÓN

Por medio del presente yo Ing. Fausto Salazar, MSc., certifico que el Sr. Roberth Gabriel Lima Carvajal portador de la cedula de ciudadanía número 2100572755, ha trabajado en el desarrollo del proyecto de grado **“Desarrollo de una aplicación móvil para reconocimiento facial con realidad aumentada utilizando Unity y AR Foundation que permita la identificación de personas dentro del campus de la Universidad Técnica del Norte.”**, previo a la obtención del Título de Ingeniero en Software realizado con interés profesional y responsabilidad que certifico con honor de verdad.

Es todo en cuanto puedo certificar a la verdad

Atentamente

Ing. Fausto Alberto Salazar Fierro, MSc.
DIRECTOR DE TRABAJO DE GRADO

DEDICATORIA

A mis queridos padres, Gloria Carvajal y Robel Lima, cuyo amor inquebrantable y esfuerzo constante han sido la base de todo lo que soy. A mi hermana, Verónica Lima, hermana y amiga incondicional y fuente de inspiración, quien con su cariño y apoyo me ha impulsado en cada paso. Ustedes son la razón detrás de cada logro, y sin su presencia, este camino habría sido imposible de recorrer. También dedico este esfuerzo a mi tía Susana Carvajal, cuyo amor y palabras siempre llegaron en los momentos más necesarios. Este trabajo es un reflejo del amor y la fortaleza que ustedes me han brindado.

Roberth Gabriel Lima Carvajal

AGRADECIMIENTO

A mi familia, a quienes debo cada paso dado en este recorrido: a mi madre, que pese a todas las circunstancias estuvo ahí para apoyarme, a mi padre que me enseñó como debe comportarse un hombre ante las adversidades y a mi hermana, que siempre supo estar conmigo en mis momentos más débiles, y que me hizo saber que nunca estaría solo y siempre podría contar con ella, al final a toda mi familia por ser mi refugio y mi motor. A mi tía Susana Carvajal, por su ayuda incondicional infinita y por siempre creer en mí. No sería quien soy sin el amor y la guía que me brindaron.

A los ingenieros Mauricio Rea, Irving Reascos, y Quiña José, cuya sabiduría y generosidad me permitieron sortear los desafíos que tuve que soportar, no solo en lo académico, si no una gran ayuda en lo personal. Su apoyo fue fundamental, y por ello, les estaré eternamente agradecido.

A mis compañeros de clase y amigos, por su constante ayuda y por los buenos momentos vividos que hicieron de este camino una interesante experiencia.

Roberth Gabriel Lima Carvajal

TABLA DE CONTENIDOS

DEDICATORIA.....	3
AGRADECIMIENTO.....	4
TABLA DE CONTENIDOS	5
ÍNDICE DE FIGURAS.....	9
ÍNDICE DE TABLAS.....	13
RESUMEN	15
ABSTRACT	16
INTRODUCCIÓN.....	17
Tema	17
Problema	17
Antecedentes	17
Situación Actual.....	17
Prospectiva.....	18
Planteamiento del problema	18
Objetivos.....	19
Objetivo General.....	19
Objetivos Específicos	20
Alcance	20
<i>Administrador Web</i>	20
<i>API REST</i>	20
<i>Aplicación móvil</i>	21
Metodología	21
Justificación	23
<i>Justificación Tecnológica</i>	24
<i>Justificación de riesgos y seguridad</i>	25
<i>Justificación Social</i>	25
<i>Justificación Educativa</i>	25

<i>Justificación Institucional</i>	25
CAPÍTULO 1	26
Marco Teórico	26
1.1. Reconocimiento facial	26
1.1.1. Introducción al reconocimiento Facial.....	26
1.1.2. Métodos y algoritmos de reconocimiento facial	28
1.1.3. Integración de Unity y AR Foundation	31
1.1.4. Aplicaciones y casos de uso del reconocimiento facial.....	33
1.2. Realidad aumentada	34
1.2.1. Concepto y definición de realidad aumentada.	34
1.2.2. Sinergia entre realidad aumentada y reconocimiento facial.	35
1.2.3. Aplicaciones y casos de uso de realidad aumentada.....	36
1.3. Metodología SCRUM	38
1.3.1. Introducción a la metodología ágil SCRUM	38
1.3.2. Procesos, artefactos y eventos en SCRUM.....	41
1.3.3. Implementación de SCRUM en el desarrollo de software.....	41
1.4. Norma ISO/IEC:25010	42
1.4.1. Concepto y objetivos de la norma ISO/IEC:25010.....	42
1.4.2. Características de calidad de software según la norma ISO/IEC:25010	44
1.4.3. Evaluación de la calidad de uso según la norma ISO/IEC:25010.	46
CAPÍTULO 2	48
Desarrollo del proyecto.....	48
2.1 Planificación.....	48
2.1.1 Historias de usuario.....	48
2.1.2 Estimación de esfuerzo	52
2.1.3 Prioridad de historias de usuario	53
2.1.4 Creación del Product Backlog	54
2.1.5 Definición de roles del proyecto.....	55

2.1.6	Planificación de Sprints	56
2.2.	Diseño.....	57
2.2.1.	Arquitectura del sistema	57
2.2.2.	Actores	58
2.2.3.	Diagramas de casos de uso	59
2.2.4.	Diagramas de Secuencia	67
2.2.5.	Diagrama de clases.....	70
2.2.6.	Prototipo del Administrador web	71
2.2.7.	Prototipo de la aplicación móvil	73
2.3.	Desarrollo del sistema.....	74
2.3.1.	Módulos del proyecto	74
2.3.2.	Sprint 0.....	75
2.3.3.	Sprint 1.....	86
2.3.4.	Sprint 2.....	92
CAPÍTULO 3	103
3.1.	Verificación de resultados	103
3.1.1.	Medición del modelo de calidad en uso.....	103
3.2.	Análisis e interpretación de resultados	105
3.2.1.	Especificación de muestra.....	105
3.2.2.	Desarrollo de encuesta.....	105
3.2.3.	Análisis general de los resultados de la encuesta	108
3.3.	Proceso de Evaluación del Modelo de Calidad en Uso	110
3.3.1.	Satisfacción del usuario.....	110
3.3.2.	Métrica de confianza	116
3.3.3.	Métrica de comodidad	122
3.4.	Resultado de la evaluación	126
3.5.	Test de Normalidad.....	128
3.6.	Correlación de Variables	129

CONCLUSIONES.....	134
RECOMENDACIONES.....	135
BIBLIOGRAFÍA	136

ÍNDICE DE FIGURAS

Figura 1 <i>Árbol de problemas</i>	19
Figura 2 <i>Gráfico con herramientas y tecnologías del proyecto</i>	21
Figura 3 <i>Metodología de desarrollo</i>	23
Figura 4 <i>Estructura del reconocimiento facial</i>	27
Figura 5 <i>Métodos de reconocimiento facial</i>	29
Figura 6 <i>Diagrama de flujo de trabajo de AR Foundation</i>	33
Figura 7 <i>Imagen de las características de la realidad aumentada</i>	35
Figura 8 <i>Flujo de trabajo SCRUM</i>	38
Figura 9 <i>Proceso iterativo de Scrum</i>	40
Figura 10 <i>Modelo de los Criterios de Calidad del Software según ISO 25010</i>	43
Figura 11 <i>Elementos que componen la ISO 25010</i>	44
Figura 12 <i>Calidad de Uso 25010</i>	46
Figura 13 <i>Satisfacción y utilidad</i>	47
Figura 14 <i>Arquitectura del sistema</i>	57
Figura 15 <i>Diagrama de caso de uso 1, Cargar imágenes y registrar datos de personas asociadas</i>	59
Figura 16 <i>Diagrama de caso de uso 2, recargar modelos de reconocimiento facial</i>	61
Figura 17 <i>Diagrama de caso de uso 3, administrar imágenes y datos</i>	62
Figura 18 <i>Diagrama de caso de uso 4, identificación en tiempo real</i>	64
Figura 19 <i>Diagrama de caso de uso 5, identificación y visualización en realidad aumentada</i>	66
Figura 20 <i>Diagrama de secuencia – Cargar imágenes de estudiantes con su información</i>	67
Figura 21 <i>Diagrama de secuencia – Recargar modelos de reconocimiento facial</i>	68
Figura 22 <i>Diagrama de secuencia – Administrar imágenes y datos</i>	68
Figura 23 <i>Diagrama de secuencia – Identificación en tiempo real</i>	69

Figura 24 <i>Diagrama de secuencia – Identificación y visualización en realidad aumentada</i>	70
Figura 25 <i>Diagrama de clases</i>	71
Figura 26 <i>Prototipo Administrador web - Inicio</i>	71
Figura 27 <i>Prototipo Administrador web - Subir</i>	72
Figura 28 <i>Prototipo Administrador web - Identificar</i>	73
Figura 29 <i>Prototipo Aplicación móvil – Screenshot 1</i>	73
Figura 30 <i>Prototipo aplicación móvil – Screenshot 2</i>	74
Figura 31 <i>Código de la página Inicio del administrador web</i>	77
Figura 32 <i>Procesamiento de las imágenes posterior a su subida</i>	78
Figura 33 <i>Base de datos alojada en Firebase (Base de datos no relacional)</i>	78
Figura 34 <i>Firebase Storage</i>	79
Figura 35 <i>Conexión a la base de datos mediante parámetros y archivo.</i>	79
Figura 36 <i>Interfaz gráfica del manejo de los datos por el lado del cliente</i>	80
Figura 37 <i>Código encargado de la subida de imágenes e información</i>	81
Figura 38 <i>Pantalla donde se deben de ingresar los datos</i>	81
Figura 39 <i>Método GET del API para recargar modelos</i>	82
Figura 40 <i>Respuesta del API al recargar modelos (Navegador)</i>	83
Figura 41 <i>Respuesta del API al recargar modelos (Consola)</i>	83
Figura 42 <i>Prueba del GET reload-models en el cliente</i>	83
Figura 43 <i>Código responsable de la eliminación de un registro</i>	84
Figura 44 <i>Pantalla de confirmación al eliminar un registro</i>	85
Figura 45 <i>Solicitud de aprobación de acceso a la cámara</i>	87
Figura 46 <i>Creación del canvas Activación de la cámara web</i>	88
Figura 47 <i>Activación de la cámara web</i>	88
Figura 48 <i>Presentación en el Administrador web del contenedor de video</i>	89
Figura 49 <i>Carga de los modelos de face-api.js</i>	89

Figura 50 <i>Verificación de identificación facial</i>	90
Figura 51 <i>Código final encargado de la detección y muestra de información</i>	91
Figura 52 <i>Conexión a la base de datos Firebase</i>	93
Figura 53 <i>Respuesta de Firebase con los datos</i>	94
Figura 54 <i>Método POST del API para subir una imagen</i>	95
Figura 55 <i>Código encargado de procesar las imágenes</i>	95
Figura 56 <i>Código encargado de consumir el POST del API (Procesar y comparar)</i>	98
Figura 57 <i>Código encargado de consumir el POST del API (Respuesta del procesado y comparativa)</i>	99
Figura 58 <i>Manejo de errores al consumir el POST del API</i>	99
Figura 59 <i>Proyecto finalizado en Unity</i>	100
Figura 60 <i>Aplicación móvil</i>	101
Figura 61 <i>Presentación de los datos con realidad aumentada</i>	101
Figura 62 <i>Matriz de calidad de Software</i>	103
Figura 63 <i>Tipo de Software</i>	104
Figura 64 <i>Característica y subcaracterística</i>	104
Figura 65 <i>Escala de valores de SUS</i>	107
Figura 66 <i>Diagrama de frecuencias del cuestionario CSUQ</i>	109
Figura 67 <i>Histograma de los puntajes CSUQ</i>	109
Figura 68 <i>Gráfico de pastel de la pregunta A</i>	112
Figura 69 <i>Gráfico de pastel de la pregunta P</i>	113
Figura 70 <i>Gráfico de pastel de la pregunta C</i>	114
Figura 71 <i>Gráfico de pastel de la pregunta O</i>	114
Figura 72 <i>Gráfico de pastel de la pregunta N</i>	115
Figura 73 <i>Gráfico de pastel de la pregunta M</i>	116
Figura 74 <i>Gráfico de pastel de la pregunta G</i>	118

Figura 75 <i>Gráfico de pastel de la pregunta H</i>	119
Figura 76 <i>Gráfico de pastel de la pregunta I</i>	119
Figura 77 <i>Gráfico de pastel de la pregunta J</i>	120
Figura 78 <i>Gráfico de pastel de la pregunta L</i>	121
Figura 79 <i>Gráfico de pastel de la pregunta K</i>	121
Figura 80 <i>Gráfico de pastel de la pregunta D</i>	123
Figura 81 <i>Gráfico de pastel de la pregunta E</i>	124
Figura 82 <i>Gráfico de pastel de la pregunta F</i>	125
Figura 83 <i>Gráfico de pastel de la pregunta B</i>	125
Figura 84 <i>Resultados de evaluación de calidad uso</i>	127
Figura 85 <i>Niveles de puntuación</i>	128
Figura 86 <i>Correlación entre las preguntas 1 y 16</i>	130
Figura 87 <i>Correlación entre las preguntas 3 y 15</i>	131
Figura 88 <i>Correlación entre las preguntas 7 y 8</i>	131
Figura 89 <i>Correlación entre las preguntas 9 y 10</i>	132
Figura 90 <i>Correlación entre las preguntas 2 y 5</i>	133

ÍNDICE DE TABLAS

Tabla 1 Avances en el reconocimiento facial	26
Tabla 2 Historia de usuario 01 – Subir imágenes de estudiantes al administrador web	48
Tabla 3 Historia de usuario 02 – Recargar los modelos de reconocimiento facial	49
Tabla 4 Historia de usuario 03 –Visualizar y administrar imágenes junto sus datos.....	50
Tabla 5 Historia de usuario 04 - Identificar una persona en tiempo real	50
Tabla 6 Historia de usuario 05 - Identificar una persona y mostrar su información en realidad aumentada.....	51
Tabla 7 Estimación de esfuerzo t-shirt sizing	53
Tabla 8 Estimación de prioridad MoSCoW.....	54
Tabla 9 Product Backlog	55
Tabla 10 Roles del proyecto.....	55
Tabla 11 Planificación de Sprints	56
Tabla 12 Actores	58
Tabla 13 Caso de uso número 1, Cargar imágenes y registrar datos de personas asociadas.	60
Tabla 14 Caso de uso número 2, recargar modelos de reconocimiento facial.....	61
Tabla 15 Caso de uso número 3, <i>administrar imágenes y datos</i>	63
Tabla 16 Caso de uso número 4, identificación en tiempo real.....	65
Tabla 17 Caso de uso número 5, identificación y visualización en realidad aumentada.....	66
Tabla 18 Módulos del proyecto	75
Tabla 19 Sprint 0.....	75
Tabla 20 Reunión de retrospectiva Sprint 0	85
Tabla 21 Sprint 1	86
Tabla 22 Reunión de retrospectiva Sprint 1	91

Tabla 23 Sprint 2.....	92
Tabla 24 Catálogo de servicios REST.....	97
Tabla 25 Reunión de retrospectiva Sprint 3	102
Tabla 26 Relación de las métricas de Satisfacción-Utilidad de la norma ISO 25010 con las preguntas de CSUQ.....	106
Tabla 27 Respuestas escala Likert de 7 puntos.....	107
Tabla 28 Tabla de frecuencia.....	108
Tabla 29 Resultado de la encuesta CSUQ respecto a la satisfacción.....	111
Tabla 30 Métrica de satisfacción	112
Tabla 31 Métrica de confianza	117
Tabla 32 Métrica de comodidad	123
Tabla 33 Resultado de la evaluación	127
Tabla 34 Resultado del test de Kolmogorov.....	129

RESUMEN

Este trabajo está compuesto por tres capítulos, donde se describe en detalle todo el proceso llevado a cabo para desarrollar el Trabajo de Grado titulado: “Desarrollo de una aplicación móvil para reconocimiento facial con realidad aumentada utilizando Unity y AR Foundation que permita la identificación de personas dentro del campus de la Universidad Técnica del Norte.”

En la introducción se abordan los antecedentes, la situación actual, la perspectiva futura, el planteamiento del problema, los objetivos generales y específicos, el alcance del proyecto y su justificación.

Para el primer capítulo, se expone el marco teórico, abarcando aspectos como el reconocimiento facial, sus métodos y aplicaciones, la realidad aumentada y su integración con el reconocimiento facial, la metodología SCRUM aplicada al proyecto, y la norma ISO/IEC:25010 para evaluar la calidad en uso del software.

Para el segundo capítulo, se describe la organización y estructura del proyecto, incluyendo las historias de usuario, la estimación de esfuerzo, la priorización, y el desarrollo del Product Backlog. Además, se detallan los roles del equipo y la planificación de los Sprints para posterior desarrollar todos los artefactos propuestos como lo son el Administrador de tareas, el API REST, y el aplicativo móvil.

Y por último en el tercer capítulo, se muestran lo que se obtuvo, con un análisis y una interpretación basada en los criterios definidos, destacando el cumplimiento de los objetivos y lo evaluado por la ISO 25010.

Finalmente se incluyen las conclusiones, recomendaciones y referencias bibliográficas.

ABSTRACT

This work is composed of three chapters, which describe in detail the entire process carried out to develop the Degree Project entitled: “Development of a mobile application for facial recognition with augmented reality using Unity and AR Foundation that allows the identification of people within the campus of the Universidad Técnica del Norte”.

The introduction covers the background, the current situation, the future perspective, the problem statement, the general and specific objectives, the scope of the project and its justification.

For the first chapter, the theoretical framework is presented, covering aspects such as facial recognition, its methods and applications, augmented reality and its integration with facial recognition, the SCRUM methodology applied to the project, and the ISO/IEC:25010 standard to evaluate the quality in the use of software.

The second chapter describes the organization and structure of the project, including user stories, effort estimation, prioritization, and the development of the Product Backlog. In addition, the roles of the team and the planning of the Sprints are detailed for the subsequent development of all the proposed artifacts such as the Task Manager, the REST API, and the mobile application.

Finally, the third chapter shows what was obtained, with an analysis and interpretation based on the defined criteria, highlighting the fulfillment of the objectives and what was evaluated by ISO 25010.

Finally, conclusions, recommendations and bibliographical references are included.

INTRODUCCIÓN

Tema

Desarrollo de una aplicación móvil para reconocimiento facial con realidad aumentada utilizando Unity y AR Foundation que permita la identificación de personas dentro del campus de la Universidad Técnica del Norte.

Problema

Antecedentes

Como menciona Maxwell (2012) experto en liderazgo, la colaboración y las conexiones entre las personas son fundamentales para el crecimiento y el progreso de una comunidad, por lo que debido a la pandemia suscitada entre los años 2020 y 2023 todo lo antes mencionado se ha visto debilitado fuertemente dentro de las comunidades humanas, y esto se puede ver evidenciado dentro de la Universidad Técnica del Norte, donde las relaciones interpersonales se han desgastado de tal manera que existe una gran ausencia de habilidades sociales, empatía, y procesos de socialización entre individuos, lo cual preocupa, ya que todo lo mencionado ayuda a sentirse más apoyado y conectado con el entorno, mejorando en todos los aspectos (World Health Organization, 2019).

Situación Actual

Para el contexto actual, se destaca la problemática relacionada con la falta de interacción social entre los estudiantes de la carrera de software. La ausencia de relaciones humanas sólidas puede tener un impacto negativo en su experiencia académica y desarrollo personal.

En la UTN, situada en la ciudad de Ibarra y con alrededor de 12.390 estudiantes matriculados según el Módulo UTN en cifras (2022), es evidente la necesidad de promover un ambiente más inclusivo y participativo, donde los estudiantes nuevos se sientan acogidos y tengan la oportunidad de establecer lazos con sus compañeros. Como lo dicen Lindsjörn et

al. (2016) esto no solo potenciará su bienestar emocional, sino que también promoverá la colaboración y el trabajo en equipo, habilidades fundamentales en la industria del software.

La seguridad dentro de la facultad también es un aspecto crucial que no se debe descuidar. En un entorno académico, es importante garantizar la protección de todos. Como afirma Schneier (2019) La seguridad debería de tenerse en cuenta como algo continuo y de varios pasos a cumplir, y no tomarse como un elemento el cual se pueda palpar, ya que no es un producto. Para ello, es necesario implementar medidas que permitan detectar posibles amenazas o comportamientos sospechosos dentro del campus.

Prospectiva

Por otro lado, la seguridad dentro de la facultad es de vital importancia y requiere una atención prioritaria. Como menciona La Asamblea Nacional (2018) dentro de la LOES del Ecuador, artículo 86, estableciendo que las instituciones de educación superior deben asegurar un entorno de aprendizaje seguro y saludable para sus estudiantes. Lo que significa que es fundamental garantizar la protección de toda la comunidad educativa, y esto implica adoptar un enfoque continuo de seguridad en lugar de considerarla como un objetivo estático. Para lo cual es necesario implementar medidas proactivas que permitan detectar y prevenir posibles amenazas o comportamientos sospechosos dentro del campus, lo cual proporcionará una mayor tranquilidad tanto a los estudiantes como al personal involucrado.

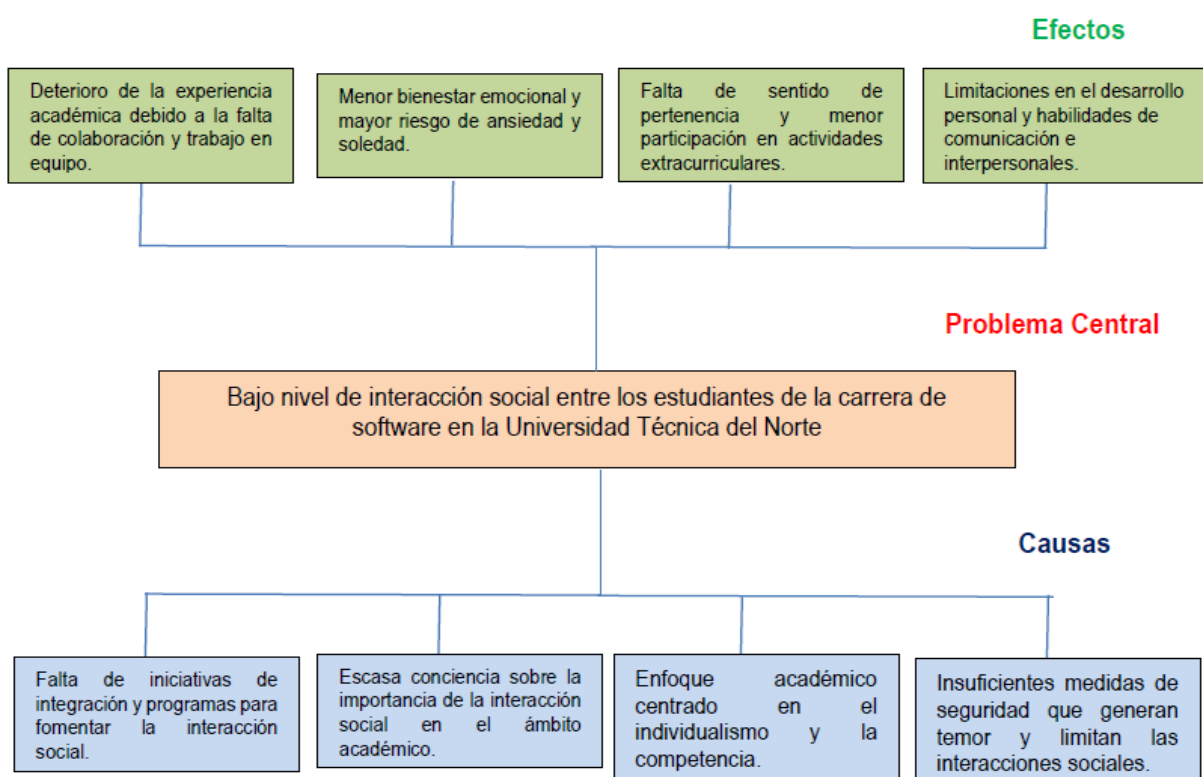
Planteamiento del problema

Es importante abordar estos desafíos sociales y de seguridad para crear un entorno propicio para el desarrollo académico y el bienestar de los estudiantes nuevos. Al promover relaciones humanas sólidas y garantizar un entorno seguro, la Universidad Técnica del Norte estará generando un efecto favorable en la experiencia educativa de sus estudiantes y en la comunidad en general.

En resumen, es fundamental prestar atención a la falta de interacción social entre los estudiantes nuevos en la carrera de software, promoviendo un ambiente inclusivo y propicio para el establecimiento de relaciones. Asimismo, se debe garantizar la seguridad dentro de la facultad, implementando medidas que detecten posibles amenazas o comportamientos sospechosos. Estos aspectos son esenciales enriquecer la experiencia educativa y promover el bienestar de los estudiantes en la Universidad Técnica del Norte.

Figura 1

Árbol de problemas



Nota. Autoría propia.

Objetivos

Objetivo General.

Desarrollar una aplicación móvil para reconocimiento facial con realidad aumentada utilizando Unity y AR Foundation que permita la identificación de personas dentro del campus de la Universidad Técnica del Norte.

Objetivos Específicos

- Establecer un marco conceptual referente al desarrollo de aplicaciones móviles con reconocimiento facial y realidad aumentada.
- Construir la aplicación móvil junto a su administrador web utilizando la metodología ágil Scrum
- Validar los resultados usando la norma ISO 25010 evaluación de calidad en uso aplicando la característica de satisfacción con la subcaracterística de utilidad.

Alcance

El presente proyecto tiene como objetivo desarrollar una app para dispositivos Android, que permita el reconocimiento, mediante un rostro, específicamente diseñada para estudiantes de primer semestre en la carrera de Software. Esta tecnología de reconocimiento facial busca personalizar la experiencia y mejorar la seguridad dentro de la Universidad Técnica del Norte, e implementando realidad aumentada, que bajo el contexto de la construcción de esta app junto al reconocimiento facial “permite a las personas dentro del campus disfrutar de un mejor aprovechamiento de un dispositivo tecnológico que poseen todos o la mayoría” (Porter & Heppelmann, 2017) la cual estará conformada por un backend, una interfaz de administración web y una app, en el que se tendrá varios módulos que en el siguiente texto se detallan.

Administrador Web

La implementación se realizará a través del empleo de tecnologías bien establecidas como Angular y Node.js, complementadas por Firebase para gestionar la base de datos y el servidor de contenidos.

API REST

La API REST se llevará a cabo utilizando Node.js y face-api.js, lo que permitirá la identificación de personas a través de cámaras en tiempo real. Esta solución integrará

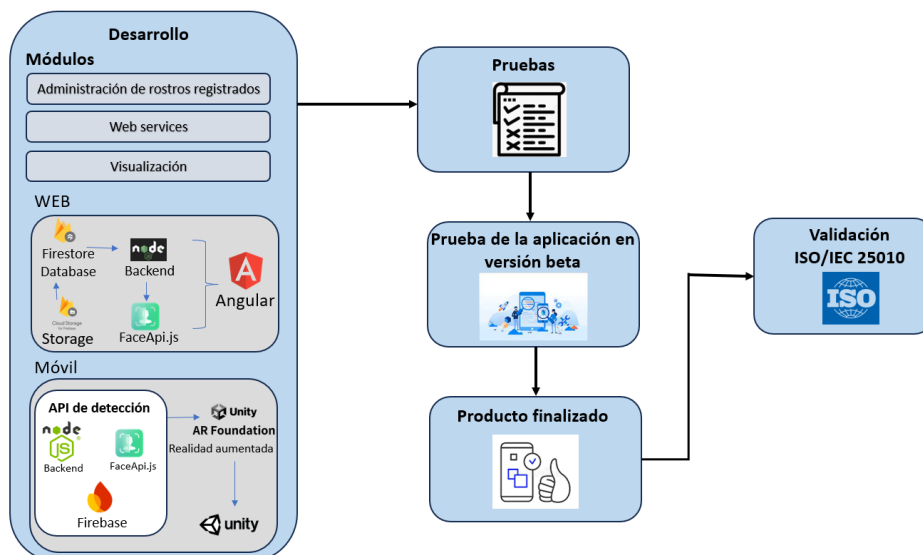
métodos HTTP para gestionar las operaciones relacionadas con los datos de las personas identificadas, facilitando así la recuperación y actualización de información en un formato JSON. Al adherirse a los principios del diseño REST, la API garantizará un manejo eficiente de las solicitudes, promoviendo un sistema sin estado que optimiza el rendimiento en aplicaciones interactivas.

Aplicación móvil

Será desarrollada para Android, para lo cual se utilizará Unity y AR Foundation, este tendrá una conexión establecida con los webs services para proporcionar toda la información requerida para su funcionamiento.

Figura 2

Gráfico con herramientas y tecnologías del proyecto



Nota. Autoría propia.

Metodología

En el primer objetivo se realizará la Revisión Sistemática de la Literatura (SLR) mediante la cual se podrá encontrar suficiente información para responder las interrogantes: ¿Cómo se puede implementar la realidad aumentada en una aplicación de reconocimiento facial? ¿De qué manera trabaja por detrás esta implementación en las aplicaciones?

El SLR emplea estrategias que minimizan los sesgos y los errores aleatorios. Estas estrategias incluyen: una búsqueda exhaustiva de todos los artículos pertinentes, criterios de selección reproducibles y claros, evaluación del diseño y las características de los estudios, así como la síntesis e interpretación de los resultados. (Manterola et al., 2013).

Se recopilarán datos provenientes de múltiples fuentes, de tal forma que se agruparán informaciones relevantes que ayuden a una comprensión más profunda de la realidad aumentada junto con el reconocimiento facial, y tópicos similares que ayuden a su entendimiento.

Para cumplir con el segundo objetivo, se hará uso de SCRUM, de forma individual, a pesar de que SCRUM está principalmente diseñado para equipos, los principios básicos también son aplicables a proyectos individuales (Scrum, 2023). De igual forma se definirán las funcionalidades claves para poder priorizarlas en el backlog de productos.

Posteriormente, se planificarán Sprints de corta duración para abordar tareas específicas durante cada iteración. Se construirá la aplicación de manera incremental, con pruebas continuas para garantizar la calidad del software. Al finalizar cada Sprint, se evaluará el progreso para poder determinar si se necesitan ajustes para mejorar. Es importante adaptar Scrum a la situación particular y llevar un seguimiento constante del avance para lograr un desarrollo exitoso (Schwaber & Sutherland, 2020).

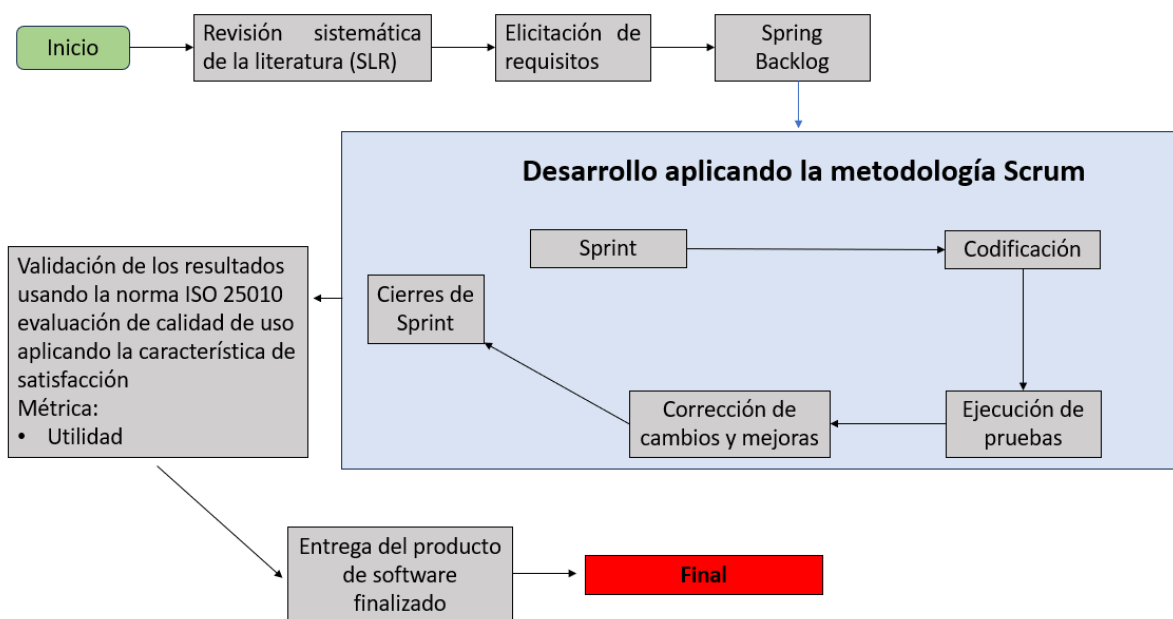
Además, con el fin de salvaguardar la privacidad de los datos y obtener el consentimiento adecuado en relación con el desarrollo de la aplicación, se implementará un proceso para solicitar una firma a través de un acta de consentimiento de información básica. Esta medida garantizará que los usuarios estén debidamente informados y consientan el uso de sus datos antes de ser publicados en la aplicación.

Y finalmente el proceso del tercer objetivo, se realizará una evaluación del producto final de software utilizando la característica de satisfacción de la ISO 25010, específicamente la subcaracterística de utilidad. Permitirá medir el grado en que el producto cumple con las

necesidades y expectativas de los usuarios. Se evaluará la utilidad del software en relación con la facilidad de uso, la eficiencia y la efectividad. La evaluación se basará en la pregunta "¿En qué medida el software cumple con las necesidades y expectativas de los usuarios?" (Pressman, 2010). La Norma proporcionará un marco sólido para evaluar la calidad de uso del producto final dentro del proceso (ver **Figura 3**).

Figura 3

Metodología de desarrollo



Nota. Autoría propia.

Justificación

Los usuarios serán estudiantes pertenecientes al primer nivel de la carrera de software. Ya que la cantidad total es sumamente basta, con un total aproximado de 12.390 estudiantes matriculados en la UTN (Universidad Técnica del Norte, 2022).

Con el presente proyecto de trabajo de grado se busca colaborar con uno de los Objetivos de Desarrollo sostenible. Objetivo 16: Un ambiente escolar inclusivo, justo y seguro pueden ser detonantes que generen mejores resultados en lo académico. Los avances

tecnológicos son de suma importancia para solucionar permanentemente a los desafíos de una sociedad más unida y segura (Naciones Unidas, 2019).

La esencia del ODS 16, de construir sociedades tranquilas y comprensivas para el progreso llevadero, está en línea con la promoción de relaciones humanas sólidas. También reconoce el valor de la participación y cooperación de las personas para establecer comunidades justas y equitativas. Al promover un ambiente donde se pueda interactuar, asistir y establecer relaciones, se contribuye a fomentar la paz y la inclusión social.

Además, el ODS 16 también destaca la importancia de instituciones sólidas y efectivas, que garanticen el amparo de los derechos y la seguridad de las personas. El problema planteado, menciona la necesidad de implementar medidas de seguridad dentro del campus universitario para detectar posibles amenazas o comportamientos sospechosos. Esto también se relaciona con el ODS 16, que busca promover instituciones sólidas y responsables que garanticen la protección y la salud de las personas.

El desarrollo de este proyecto se relaciona con la visión de la Carrera de Software, mediante la cual expresa que forma ingenieros capacitados, críticos, con enfoque humanista, liderazgo y espíritu emprendedor, comprometidos con la responsabilidad social; impulsa, promueve y lleva a cabo procesos tecnológicos, científicos e innovadores en la gestión y desarrollo de software de calidad (Carrera de Software, 2020).

Justificación Tecnológica

La creación de una aplicación móvil que incorpora reconocimiento facial y realidad aumentada utilizando Unity y AR Foundation permitirá aprovechar las capacidades tecnológicas actuales para mejorar la seguridad y la identificación de personas dentro del

campus. Aumentando así, el conocimiento acerca del desarrollo e implementación de RA, ya que no se lo tiene muy familiarizado.

Justificación de riesgos y seguridad

La adopción de esta aplicación constituye una medida esencial para fortalecer la seguridad y salvaguardar a los usuarios dentro de la UTN. Al permitir una identificación ágil de individuos dentro del campus, se agilizarán las respuestas ante situaciones de emergencia y mejorará la coordinación de acciones en situaciones de catástrofes naturales u otros sucesos críticos. Asimismo, esta herramienta brindaría mayor tranquilidad a estudiantes, profesores y personal, generando un entorno más seguro y confiable, fomentando así la confianza mutua y el amparo de los involucrados.

Justificación Social

La implementación de esta aplicación fomentará un ambiente más seguro y confiable, generando confianza entre los usuarios, promoviendo la protección de sus derechos y bienestar.

Justificación Educativa

Este proyecto mejorará la socialización de los nuevos estudiantes, sintiendo que poseen una forma más interactiva de poder reconocerse los unos a los otros, e inclusive sintiéndose más seguros, tanto social, como de forma íntegra.

Justificación Institucional

El desarrollo de esta aplicación se alinea con los objetivos y principios de la Universidad Técnica del Norte al promover un ambiente seguro y propicio para el aprendizaje. Además, responde a las demandas actuales de la sociedad y contribuye a la imagen institucional y la excelencia académica.

CAPÍTULO 1

Marco Teórico

1.1. Reconocimiento facial

1.1.1. Introducción al reconocimiento Facial.

En la **Tabla 1**, se menciona que esta tecnología tiene una historia que se remonta varias décadas atrás, con avances significativos a lo largo del tiempo. Desde el desarrollo inicial de métodos semiautomáticos para ingresar medidas faciales en las computadoras hasta la introducción posterior de la IA, que ayuda a la interpretación de imágenes, cada etapa ha contribuido al progreso de esta tecnología. Un ejemplo de ellos es la presentación de Eigenfaces, un método que emplea el Análisis de Componentes Principales (PCA), lo que marcó un hito clave en este campo. Estos avances impulsaron el uso de esta técnica como una herramienta fundamental en áreas como la visión por computadora y la biometría (Adjabi et al., 2020).

Tabla 1

Avances en el reconocimiento facial

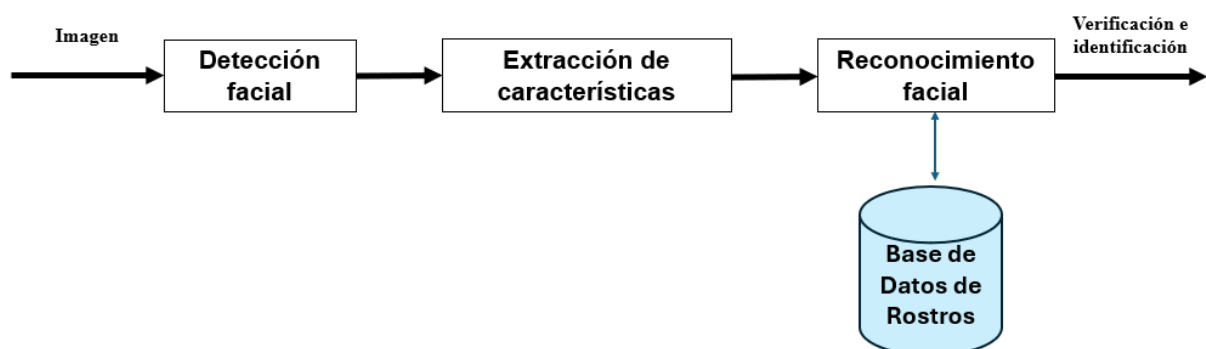
Año	Avance	Investigador(es) y Contribuciones
1964	Desarrollo de método semiautomático para ingresar medidas faciales en computadora	Investigadores estadounidenses, liderados por Bledsoe, proponen un método semiautomático de ingreso de medidas faciales en computadora.
1977	Mejora del sistema agregando 21 marcadores adicionales	Investigadores continúan desarrollando el sistema, agregando 21 marcadores adicionales como el ancho de los labios y el color del cabello.
1988	Introducción de la inteligencia artificial para interpretar imágenes de manera más eficiente	La inteligencia artificial es incorporada para interpretar imágenes de forma más eficiente, utilizando matemáticas como el álgebra lineal.
1991	Presentación de Eigenfaces utilizando Análisis de Componentes Principales (PCA)	Alex Pentland y Matthew Turk del MIT presentan Eigenfaces, que utiliza el método estadístico de Análisis de Componentes Principales (PCA) para el reconocimiento facial.

Nota. Adaptado de (Adjabi et al., 2020).

Según Venturini et al. (2021) el reconocimiento facial (FR) ha avanzado gracias a aplicaciones como seguridad y publicidad, impulsadas por tecnologías como el aprendizaje profundo. Y dicha tecnología puede entenderse en dos sentidos, como únicamente el reconocimiento de un rostro y su forma, o como los dos procesos de reconocer un rostro e identificarlo (ver **Figura 4**). Aunque nació en los 60, recientes avances en captura y procesamiento de imágenes han sido clave. Estos permiten entrenar algoritmos con grandes conjuntos de datos para adaptarse a diversos entornos, mejorando la calidad de las capturas y precisión en la identificación. Este progreso ha sido fundamental en sectores como seguridad, vigilancia, publicidad personalizada, asistencia médica y la investigación académica.

Figura 4

Estructura del reconocimiento facial



Nota. Adaptado de (Kortli et al., 2020).

Con todo lo mencionado previamente y como se muestra en la **Figura 4**, se puede dejar claro que esto implica la identificación de personas a través de rasgos distintivos de sus rostros. Este método incluye la captura y análisis de estos patrones faciales, que luego se cotejan con la base de datos para conocer la identidad del individuo (Hernández, 2010).

Hernández también menciona que dicha tecnología posee ciertos componentes claves que aseguran en su mayor medida un correcto FR, algunos de estos son:

- Captura de imagen
- Preprocesamiento de imagen

- Extracción de características
- Comparación y coincidencias
- Técnicas de reconocimiento

Estos trabajan en conjunto para lograr la identificación precisa de individuos a partir de sus características faciales únicas.

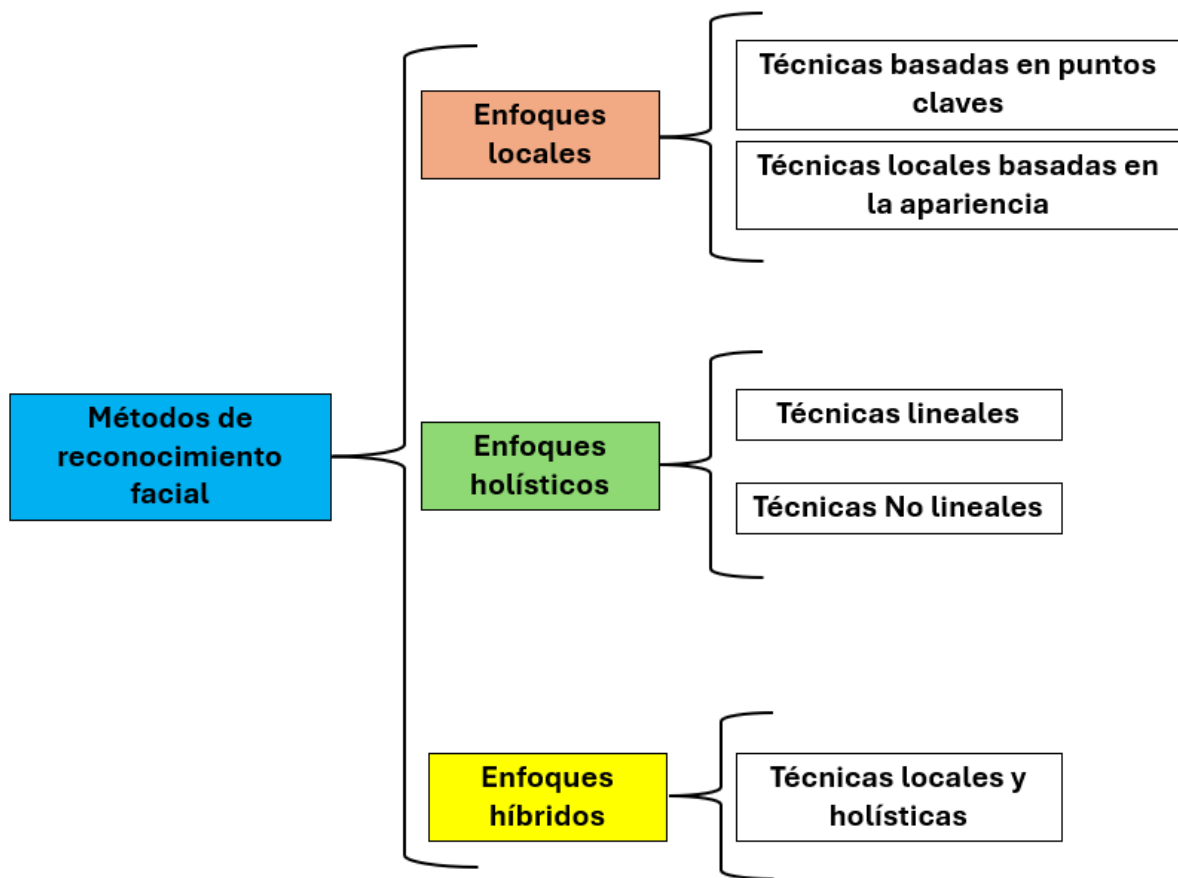
También emplea tecnologías como algoritmos neuronales para detectar características faciales, aquí se puede mencionar algunas aplicaciones comunes como la verificación de identidad y análisis emocional. De igual manera sectores como animación, seguridad y dispositivos móviles lo utilizan ampliamente debido a que los beneficios incluyen precisión y detección de emociones, pero enfrenta desafíos en tiempo real y rendimiento en dispositivos móviles. Para ello es crucial mejorar la adaptabilidad y precisión en condiciones difíciles (Khabarлак & Koriashkina, 2021).

1.1.2. Métodos y algoritmos de reconocimiento facial

El FR es una rama de la biometría que se centra en detectar o corroborar la coincidencia de una persona utilizando una imagen. Los sistemas que lo poseen usan tecnologías de visión por computadora y machine learning, con el objetivo de identificar rasgos distintivos en las imágenes de los rostros y luego compararlas con las almacenadas en una database. Métodos que se explican en la **Figura 5**. Las mejoras logradas en estas tecnologías han permitido el desarrollo de sistemas de FR más precisos y eficientes (Navarro et al., 2022).

Figura 5

Métodos de reconocimiento facial



Nota. Adaptado de (Kortli et al., 2020).

Dentro del documento se van a hablar de 2 tipos de reconocimiento.

Reconocimiento basado en características

El FR basado en características, también conocido como geométrico, se basa en la medición y comparación de los aspectos físicos del rostro. Estas pueden incluir la separación ocular, longitud nasal, forma de los malaras, entre otros. Estos sistemas emplean algoritmos para extraer las características de las imágenes faciales y, posteriormente, las confrontan con las almacenadas en un repositorio de datos para reconocer o verificar la identidad de una persona (Arguello Fuentes, 2011).

Pese a su simplicidad y eficacia en ciertas aplicaciones, esta forma de hacerlo también tiene sus limitaciones. Por ejemplo, estos sistemas pueden verse afectados por cambios en la iluminación, la pose y la expresión facial. Además, la exactitud de la respuesta puede verse afectada por la calidad de las imágenes faciales y la precisión de la extracción de características (Cadena et al., 2017).

Para superar estas limitaciones, se inventaron técnicas más avanzadas, por ejemplo, el uso de modelos 3D y tecnologías de aprendizaje profundo. Estas técnicas permiten una representación más precisa y robusta de las características faciales, lo que puede mejorar la precisión y la robustez de los sistemas de FR (Arguello Fuentes, 2011).

Reconocimiento basado en redes neuronales

El FR que se basa en redes neuronales hace uso de modelos artificiales para aprender y extraer características faciales. Estos sistemas pueden ser entrenados con un gran número de imágenes, lo que les permite identificar automáticamente los rasgos más relevantes para el reconocimiento. Esto resulta en un desempeño más preciso y robusto en comparación con los enfoques tradicionales (Jiménez et al., 2018).

Las redes neuronales convolucionales (CNN) son las más utilizadas en el FR por su eficacia en el análisis de fotografías. Estas redes aprenden a identificar y extraer atributos de alto nivel de los rostros, como la forma de los rasgos y las texturas de la piel, superando a los métodos tradicionales basados en características (Jiménez et al., 2018).

A pesar de sus ventajas, la usar dicho método también presenta varios desafíos. Por ejemplo, su entrenamiento necesita una enorme cifra de información y capacidad física computacional. Además, estas redes pueden ser sensibles a los cambios en la iluminación, la pose y la expresión facial, aunque se han desarrollado técnicas para mitigar estos problemas (Mateus et al., 2014).

Para este proyecto de desarrollo se utilizó fase-api.js, framework de JavaScript que hace posible la implementación de sistemas de FR en el navegador y en Node.js. Esta biblioteca, creada por Vincent Mühler, se basa en la API central de tensorflow.js y ofrece una variedad de funcionalidades, incluyendo la detección de rostros, su reconocimiento, la expresión, la edad y la detección de género (Mühler, 2020).

Utiliza modelos de aprendizaje profundo para realizar sus tareas de detección y FR. Por ejemplo, para la detección de rostros, implementa un SSD (Single Shot Multibox Detector) basado en MobileNetV11. Este modelo de red neuronal calcula las ubicaciones de cada rostro en una imagen y devuelve los cuadros delimitadores junto con su probabilidad para cada rostro. Además, también utiliza una versión aún más pequeña de Tiny Yolo V2, lo que lo hace extremadamente amigable para dispositivos móviles y clientes con recursos limitados (Ichi, 2020).

1.1.3. Integración de Unity y AR Foundation

Unity es una plataforma para la creación de videojuegos ampliamente utilizado que soporta la creación de interacciones en entornos 2D y 3D. Mientras que AR Foundation es un marco de trabajo proporcionado por Unity que simplifica la creación de aplicaciones de realidad aumentada (AR) y que a su vez también es multiplataforma. Este marco combina las capacidades de ARKit de Apple y ARCore de Google, permitiendo el desarrollo de apps AR funcionen en un sinnúmero de celulares (Unity Technologies, 2024a).

Al enfocar este motor desarrollo a un ámbito de FR, este se vuelve sumamente útil, debido a que es una tecnología que permite identificar o verificar a un individuo a partir de un frame o varios. En el contexto de las aplicaciones móviles, esta tecnología sirve para un sinnúmero de propósitos, como la autenticación de usuarios y el desarrollo de experiencias interactivas. Que junto a la integración de AR Foundation con Unity permite aprovechar las capacidades de FR de los dispositivos móviles para crear aplicaciones innovadoras (Busto, 2023).

La combinación antes mencionada ofrece varios beneficios para el desarrollo de aplicaciones de FR, esto se debe a que uno proporciona un entorno de desarrollo robusto y flexible, mientras que el otro simplifica la implementación de características AR avanzadas. Esta integración permite crear aplicaciones que no solo son funcionales, sino también altamente interactivas y atractivas para los usuarios (Unity Technologies, 2024b).

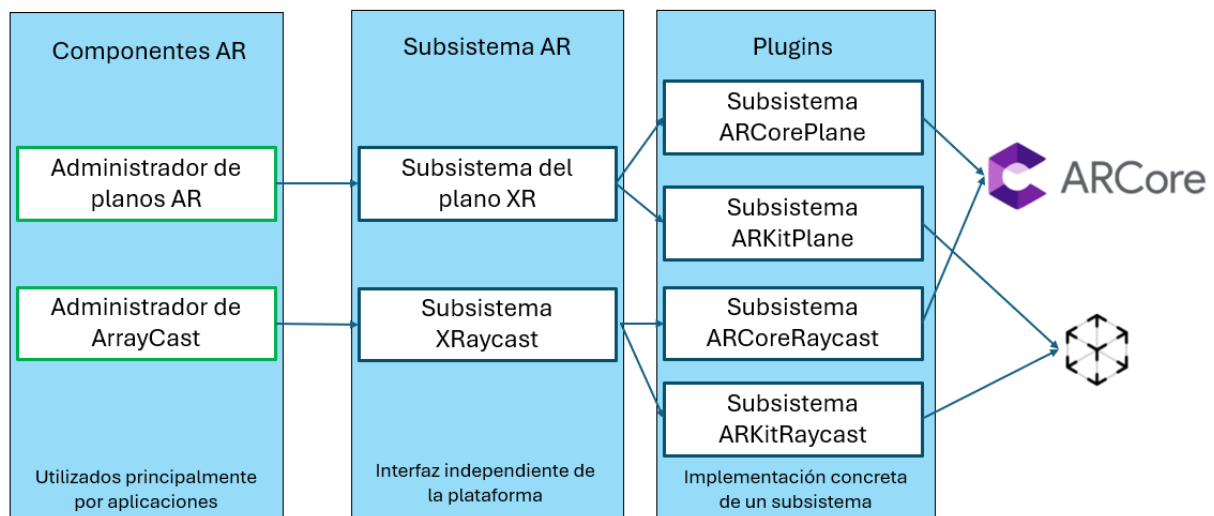
Para implementar el FR utilizando ambas tecnologías, se debe configurar los proyectos de tal forma que puedan utilizar ARKit o ARCore (ver **Figura 6**) dependiendo de la plataforma objetivo. AR Foundation proporciona una API unificada que optimiza la localización y seguimiento de rostros en tiempo real. Dentro de Unity se puede utilizar el API de AR para crear aplicaciones que respondan dinámicamente a las expresiones faciales y movimientos del usuario (Gyata, 2024).

Aunque esta integración ofrece muchas ventajas, también presenta desafíos, como lo es garantizar la precisión y la velocidad del FR en tiempo real. Además, se debe considerar la confidencialidad y protección de la información faciales. Es crucial implementar medidas adecuadas para proteger esta información sensible (Chan, 2022).

Las aplicaciones que dan como producto esta integración que tienen como objetivo el FR son vastas y variadas. Desde aplicaciones de entretenimiento y juegos hasta herramientas educativas y de salud, las posibilidades son infinitas. Con el continuo progreso de la tecnología, es probable que surjan más innovaciones en este ámbito, lo que brindará nuevas oportunidades a los desarrolladores que se centran en estas tecnologías, crear experiencias aún más inmersivas y personalizadas (Brohm et al., 2017).

Figura 6

Diagrama de flujo de trabajo de AR Foundation



Nota. Adaptado de (Jakl, 2021).

1.1.4. Aplicaciones y casos de uso del reconocimiento facial

El FR ha evolucionado en una herramienta de gran valor para la seguridad y la inspección de acceso en diversas instituciones y entornos. Por ejemplo, en el ámbito universitario, se ha utilizado para garantizar la integridad de los exámenes realizados de forma remota mientras duraba la pandemia. En este escenario, esta tecnología ofrece la capacidad de validar la correcta y verifica identidad el estudiante mostrado frente a la cámara, y prevenir el fraude académico. Sin embargo, el uso de estas también afronta retos con respecto a la seguridad de privacidad y ética, lo que requiere un enfoque cauteloso y considerado (Aznarte et al., 2022).

Además, el FR se utiliza en sistemas de control de acceso para proporcionar una capa adicional de seguridad. Estos sistemas pueden ser utilizados en una variedad de contextos, desde edificios de oficinas hasta instalaciones gubernamentales, y pueden ayudar a prevenir el acceso no autorizado. Sin embargo, al igual que con otras aplicaciones, el uso de estas tecnologías para el control de acceso plantea desafíos en términos de privacidad y ética (Lorena et al., 2022).

La identificación biométrica es otro campo importante de aplicación. Las tecnologías de FR permiten la identificación o verificación de individuos basándose en ciertos rasgos característicos del rostro. Estas tecnologías pueden ser aplicadas a una diversidad de situaciones, tales como la verificación de identidad en transacciones bancarias (Jaramillo, 2021).

Además, proporciona una forma segura y eficiente de verificar la identidad de una persona. Estos sistemas pueden ser utilizados en una variedad de contextos, desde la seguridad de las fronteras hasta la prevención del fraude. Sin embargo, al igual que con otras aplicaciones, y como se ha mencionado ya anteriormente, el uso de estas tecnologías para la identificación biométrica plantea desafíos en términos de privacidad y ética (idrd, 2020).

Esto se está utilizando cada vez más en aplicaciones móviles y web para una variedad de propósitos, desde la autenticación de usuarios hasta la adaptación de la interacción del usuario. Por ejemplo, algunas aplicaciones lo utilizan para proporcionar recomendaciones personalizadas basadas en las emociones del usuario. Sin embargo, nuevamente se hace énfasis en que el uso de estas tecnologías, que ahora refiere en el ámbito móvil y web, plantea desafíos en lo que respecta a la privacidad y seguridad de la información (Cabanillas, 2022).

1.2. Realidad aumentada

1.2.1. Concepto y definición de realidad aumentada.

La AR es una herramienta tecnológica que integra información digital con el mundo físico, mejorando la experiencia del usuario en relación con su entorno y cómo interactúa con él. Esta se basa en la combinación de elementos físicos y virtuales en tiempo real, dentro de smartphones, tablets y/o lentes de RA (Azuma, 1997).

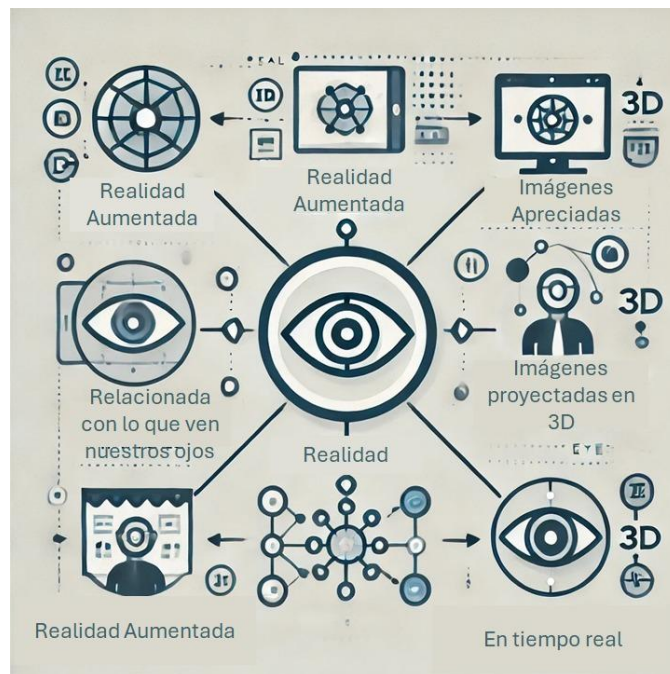
Su definición ha evolucionado con el tiempo, pero generalmente se refiere a la capacidad de integrar información digital en el entorno físico de manera que parezca coexistir con él. Esta integración se logra mediante el uso de tecnologías de seguimiento y

reconocimiento de objetos, que permiten al sistema identificar y superponer información relevante sobre los objetos del mundo real (Billinghurst et al., 2014).

A diferencia de la realidad virtual, la AR no produce un ambiente enteramente artificial, sino que añade elementos digitales al entorno existente (Ver **Figura 7**). Esto posibilita que los usuarios interactúen con el entorno real mientras obtienen información adicional, que puede mejorar su experiencia y comprensión de su alrededor (Carmigniani et al., 2011).

Figura 7

Imagen de las características de la realidad aumentada



Nota. (ChatGPT, 2024)

1.2.2. Sinergia entre realidad aumentada y reconocimiento facial.

La sinergia entre la AR y el FR se está convirtiendo en tema de interés, que aumenta en el campo de la tecnología. La integración de estas dos promete crear experiencias interactivas y personalizadas que pueden mejorar la manera en que interactuamos con el entorno virtual. Por ejemplo, las aplicaciones de AR pueden utilizar el FR para identificar a los usuarios y adaptar el contenido en función de su identidad.

La convergencia de ambas se manifiesta en una variedad de aplicaciones. Por ejemplo, en el ámbito de la seguridad, las aplicaciones de AR la utilizan para autenticar la identidad antes de brindarles acceso a los datos. En el ámbito educativo, las aplicaciones pueden utilizar la identificación facial para ajustar la experiencia educativa de los estudiantes, modificando el contenido en función de su progreso y rendimiento.

La factibilidad de la sinergia se ve facilitada por los avances en ambas. El FR se ha vuelto más preciso y eficiente gracias a los avances en la interpretación visual computarizada y las técnicas de aprendizaje inteligente. Del mismo modo, la AR se ha vuelto más inmersiva y realista gracias a las mejoras en los gráficos por computadora y las tecnologías de seguimiento. Estos avances han hecho posible la integración efectiva de ambas, en una amplia gama de aplicaciones (Aznarte et al., 2022).

Las aplicaciones son diversas y están en constante crecimiento. Por ejemplo, en el ámbito del comercio minorista, las tiendas pueden utilizar la AR y el FR para ofrecer experiencias de compra personalizadas. Los clientes pueden utilizar sus dispositivos móviles para sobreponer los datos de productos en su entorno real, y la tienda puede utilizar la identificación facial para identificar a los clientes y ofrecer recomendaciones personalizadas (Venturini et al., 2021).

A pesar de las prometedoras aplicaciones que surgen al juntar ambas tecnologías, también existen desafíos. Como la privacidad de la información. El FR implica la recopilación y proceso de la Información biométrica, lo que provoca dudas sobre la privacidad y la integridad de los datos. Además, la AR puede requerir el acceso a la cámara y la ubicación del dispositivo del usuario, lo que también puede plantear preocupaciones de privacidad (Venturini et al., 2021).

1.2.3. Aplicaciones y casos de uso de realidad aumentada

La aplicación abarca diversos campos, desde el ámbito educativo y de entretenimiento hasta el sector médico e industrial. En la educación, por ejemplo, la AR puede proporcionar

aprendizajes más dinámicos e interactivos y envolventes, posibilitando a los estudiantes visualizar conocimientos complejos de forma entendible (Wu et al., 2013).

Además, tiene un gran potencial en el ámbito de la medicina, donde puede ser utilizada para mejorar la precisión de procedimientos quirúrgicos, proporcionar formación médica avanzada y ofrecer nuevas formas de rehabilitación para los pacientes. La capacidad de sobreponer datos en el cuerpo humano de manera instantánea puede ayudar a los médicos a tomar decisiones más acertadas y precisas (Muensterer et al., 2014).

También se está convirtiendo en una herramienta valiosa en los sectores educativos y de formación. Dado que puede fomentar una mayor interacción y compromiso de los estudiantes, proporcionando experiencias de aprendizaje inmersivas y contextualizadas. Igualmente, puede facilitar el entendimiento de ideas complejas, permitiendo que los estudiantes interactúen y manipulen objetos virtuales en un entorno tridimensional (Fernández de Terán, 2019).

Del mismo modo en el lado del entretenimiento y los videojuegos, la AR está creando nuevas formas de interacción y compromiso. Los videojuegos que la usan, como Pokémon Go, han demostrado el potencial de esta, para crear experiencias de juego inmersivas y atractivas. Además, también está siendo utilizada en el cine y la televisión para mejorar la experiencia de visualización al superponer información y gráficos en las imágenes en tiempo real (Perdomo et al., 2020).

La RA está transformando la publicidad y el marketing al permitir a las marcas interactuar con los consumidores de formas nuevas y emocionantes. Las campañas de publicidad que emplean esta estrategia pueden ofrecer experiencias interactivas que aumentan el compromiso del consumidor y mejoran el recuerdo de la marca. Además de proporcionar a los clientes la oportunidad de explorar productos a través de medios virtuales antes de comprarlos, lo que puede influir positivamente en las decisiones de compra (Cárdenas et al., 2018).

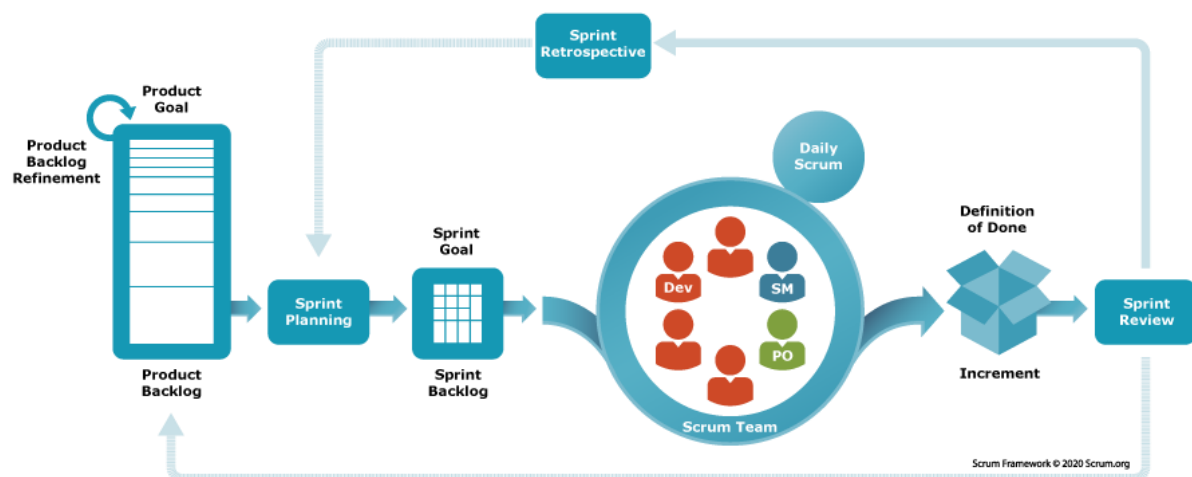
1.3. Metodología SCRUM

1.3.1. Introducción a la metodología ágil SCRUM

SCRUM (del inglés "Scrum", que no es un acrónimo, pero hace referencia a una formación en rugby) es una metodología ágil para administración de proyectos, particularmente en la creación de software. Se fundamenta en un conjunto de elementos y prácticas que siembran la colaboración, la flexibilidad y la revisión continua de avance. Proporciona un cuadro de trabajo que ofrece al equipo trabajar de manera iterativa e incremental, adaptándose rápidamente a los cambios y entregando productos de alta calidad (Estrada et al., 2021).

Figura 8

Flujo de trabajo SCRUM



Nota. (Scrum, 2023).

Se caracteriza por su enfoque en la autoorganización y la responsabilidad compartida. Los equipos son autónomos y tienen la libre potestad de gestionar sus decisiones acerca del camino de su trabajo, como lo muestra la **Figura 8**. Esto incentiva la colaboración, la comunicación y el avance continuo, lo que resulta en equipos más eficientes y productos de mayor calidad (Estrada et al., 2021).

Además, SCRUM promueve la transparencia y la inspección frecuente del avance del proyecto. Esto logra que identificar, resolver problemas rápidamente, y adaptarse a las modificaciones en las especificaciones o en el contexto del proyecto. Como resultado, ayuda a los equipos a proveer soluciones que cumplen con las expectativas del cliente y aportan valor al negocio (Estrada et al., 2021).

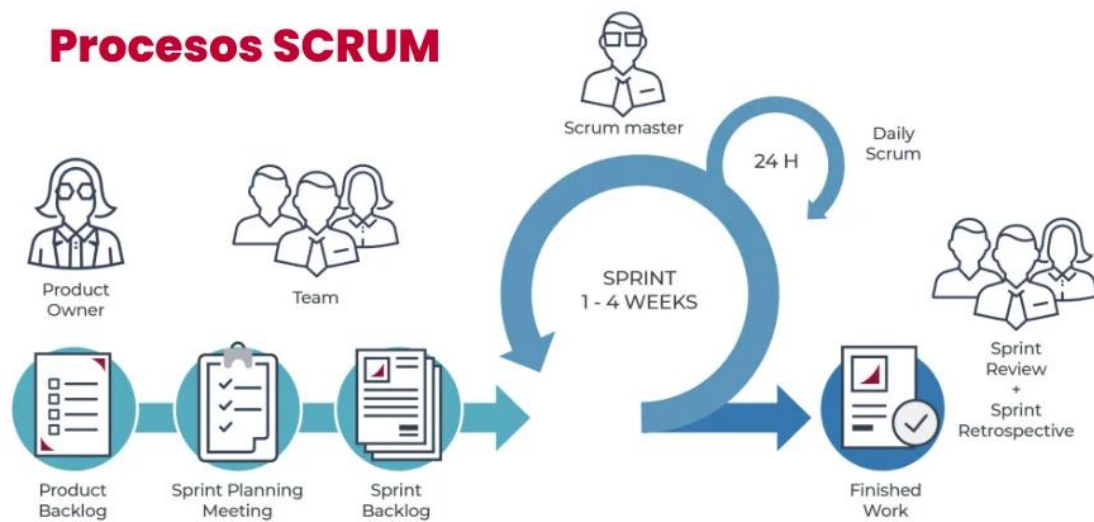
Esta metodología cuenta con principios y valores ágiles, los cuales son fundamentales para sí misma. Estos incluyen la asistencia, la revisión de valor, la adaptación y el respeto, las cuales guían las decisiones y acciones de los equipos. Al adherirse a estos fundamentos, los equipos pueden trabajar de manera más eficaz y eficiente, dando como resultado una entrega de artículos de calidad superior que satisfacen las necesidades del cliente (Mancuzo, 2021).

Asimismo, los principios y valores ágiles fomentan un enfoque constante en la mejora progresiva. Los grupos de trabajo están infatigablemente buscando formas de optimar sus procesos y prácticas, que por consiguiente puedan entregar más valor a sus clientes. Esta cultura de mejora continua es fundamental para lograr el éxito en cualquier proyecto que utilice esta metodología ágil (Mancuzo, 2021).

En la **Figura 9** se pueden apreciar los principales 3 roles: el Product Owner, el Scrum Master y el Equipo de Desarrollo. Cada puesto implica responsabilidades concretas y juega un papel crucial en el éxito del proyecto (Alaimo, 2018).

Figura 9

Proceso iterativo de Scrum



Nota. (Araneda, 2022).

El Product Owner tiene la responsabilidad de optimizar el valor del producto y administrar el Product Backlog. Este rol demanda un conocimiento profundo tanto de las necesidades del cliente como del negocio, además de la habilidad para establecer las prioridades del proyecto.

El Scrum Master, por otro lado, es responsable de velar por el cumplimiento de los principios y metodologías de SCRUM por parte del equipo. Este rol asiste al equipo a resolver impedimentos, facilita las reuniones y trabaja para establecer un ambiente que permita al equipo trabajar de manera eficiente.

Finalmente, el Equipo de Desarrollo se encarga de entregar incrementos de producto que puedan ser entregados al finalizar cada Sprint. Los miembros aquí deben tener habilidades, conocimientos técnicos, y deben ser capaces de trabajar juntos de manera efectiva para entregar productos de alta calidad.

1.3.2. Procesos, artefactos y eventos en SCRUM

SCRUM establece diversos eventos que se desarrollan a lo largo de cada sprint: Sprint Planning, Daily Scrum, Sprint Review y Sprint Retrospective. En el Sprint Planning, el equipo se reúne para colaborar y discutir las tareas a realizar durante el sprint, además de definir el objetivo de este. El Daily Scrum es una breve reunión de unos 15 minutos, en la que el equipo evalúa su progreso hacia el cumplimiento del objetivo del sprint. (Deloitte, 2024).

El Sprint Review se centra en el producto desarrollado, específicamente en el avance del producto potencialmente entregable planificado en el sprint. Finalmente, la Sprint Retrospective es la posibilidad para que el equipo se analice y proponga mejoras para que el desarrollo del siguiente Sprint sea más eficiente (Deloitte, 2024).

El Product Backlog es un índice priorizado de compromiso para los desarrolladores que se obtiene de la planificación y las especificaciones del producto (Radigan, 2021).

El Sprint Backlog consiste en una lista de elementos del Product Backlog que el equipo se compromete a finalizar en un sprint determinado. El incremento es el resultado de todo el trabajo realizado durante el sprint, el cual se agrega a los incrementos previos (Arroyo, 2023).

Los eventos, o ceremonias, son momentos específicos dentro del tiempo en los que se realizan actividades determinadas. Estos eventos ofrecen oportunidades para la inspección y la adaptación continua, lo cual es clave para preservar la agilidad en el proceso. Entre ellos se incluyen el Sprint, Sprint Planning, Daily Scrum, Sprint Review y Sprint Retrospective (Lance, 2023).

1.3.3. Implementación de SCRUM en el desarrollo de software.

SCRUM es un enfoque ágil que se fundamenta en un conjunto de prácticas y métodos para la administración de proyectos. Estas están diseñadas para promover el trabajo en equipo, la adaptabilidad y la entrega constante de valor. Algunas prácticas claves incluyen el

uso de Sprints, Sesiones diarias de SCRUM, revisiones de los resultados del sprint y retrospectivas (Schwaber & Sutherland, 2020).

Además, promueve la transparencia y la inspección frecuente del avance del proyecto. Esto aprueba a los equipos identificar y resolver problemas rápidamente, y adaptarse a las modificaciones en los requisitos o en las condiciones del proyecto. Como resultado, ayuda a los equipos a proporcionar productos que responden a las necesidades del cliente y aportan valor (Productiviza, 2024).

Existen varias herramientas y software disponibles para la dirección de proyectos ágiles. Las cuales proporcionan funciones que facilitan la implementación de las prácticas y técnicas de SCRUM. Entre las más conocidas se encuentran Jira, Trello, Asana, y Microsoft Teams (Sobrevilla et al., 2017).

Así mismo, proporcionan características como La organización de tareas, el monitoreo del tiempo, el trabajo en equipo, y la integración con otras herramientas. Estas características permiten a los equipos encargarse eficazmente sus proyectos, colaborar de forma adecuada, y entregar productos de alta calidad (Camilo et al., 2021).

1.4. Norma ISO/IEC:25010

1.4.1. Concepto y objetivos de la norma ISO/IEC:25010

La norma ISO/IEC 25010 proporciona un marco de referencia para valorar la calidad de los sistemas. Tiene como objetivo principal es aumentar la satisfacción del usuario, mejorar la productividad y reducir los gastos de desarrollo. Para lograr esto, determina los requisitos y los estándares de evaluación (Ver **Figura 10**) que facilitan la evaluación de la calidad de un producto de software (Arias-Orezano et al., 2021).

Figura 10

Modelo de los Criterios de Calidad del Software según ISO 25010



Nota. Adaptado de (Ormeño, 2019).

La norma tiene un alcance extenso y abarca diversos contextos, como el desarrollo, la gestión y la adquisición de software. Es relevante para cualquier organización o individuo involucrado en lo antes mencionado o en la implementación de productos de software (Arias-Orezano et al., 2021).

La estructura se basa en un modelo de calidad que representa el pilar del sistema para la evaluación de la calidad del producto. Sistema para medir la calidad del producto. Este modelo establece las características de calidad que se considerarán al evaluar las propiedades de un producto de software (ISO 25010, 2022).

Se establece que los atributos de calidad principales incluyen ocho categorías: funcionalidad, eficiencia de desempeño, compatibilidad, usabilidad, fiabilidad, seguridad,

facilidad de mantenimiento y adaptabilidad. Cada uno de estos se subdivide en varias subcaracterísticas, que proporcionan una descripción más detallada de cada carácter de calidad (ISO 25010, 2022).

En el contexto de la satisfacción del usuario y la utilidad, este estándar considera la satisfacción como una de las subcaracterísticas de la usabilidad. La utilidad, por otro lado, se considera como parte de la adecuación funcional, que es una subcaracterística de la funcionalidad (Rocha, 2020).

1.4.2. Características de calidad de software según la norma ISO/IEC:25010

a norma ISO/IEC 25010 es una herramienta clave para valorar la calidad del software, como se indica en la **Figura 11**, se divide en nueve características principales y varias Subcaracterísticas, lo cual permite asegurar que los productos de software cumplan con las demandas explícitas e implícitas de los usuarios y otras partes interesadas, este modelo proporciona un marco para especificar, evaluar y medir la calidad del software de forma ordenada (ISO 25010, 2021).

Figura 11

Elementos que componen la ISO 25010

CALIDAD DEL PRODUCTO SOFTWARE								
ADECUACIÓN FUNCIONAL	EFICIENCIA DE DESEMPEÑO	COMPATIBILIDAD	CAPACIDAD DE INTERACCIÓN	FIABILIDAD	SEGURIDAD	MANTENIBILIDAD	FLEXIBILIDAD	PROTECCIÓN
COMPLETITUD FUNCIONAL	COMPORTAMIENTO TEMPORAL	COEXISTENCIA	RECONOCIBILIDAD DE ADECUACIÓN	AUSENCIA DE FALLOS	CONFIDENCIALIDAD	MODULARIDAD	ADAPTABILIDAD	RESTRICCIÓN OPERATIVA
CORRECCIÓN FUNCIONAL	UTILIZACIÓN DE RECURSOS	INTEROPERABILIDAD	APRENDIZABILIDAD	DISPONIBILIDAD	INTEGRIDAD	REUSABILIDAD	ESCALABILIDAD	IDENTIFICACIÓN DE RIESGOS
PERTINENCIA FUNCIONAL	CAPACIDAD		OPERABILIDAD	TOLERANCIA A FALLOS	NO-REPUDIO	ANALIZABILIDAD	INSTALABILIDAD	PROTECCIÓN ANTE FALLOS
			PROTECCIÓN FRENTE A ERRORES DE USUARIO	RECUPERABILIDAD	RESPONSABILIDAD	CAPACIDAD DE SER MODIFICADO	REEMPLAZABILIDAD	ADVERTENCIA DE PELIGRO
			INVOLUCRACIÓN DEL USUARIO		AUTENTICIDAD	CAPACIDAD DE SER PROBADO		INTEGRACIÓN SEGURA
			INCLUSIVIDAD		RESISTENCIA			
			ASISTENCIA AL USUARIO					
			AUTO-DESCRIPTIVIDAD					

Nota. (ISO 25010, 2021).

Una de las características clave de esta norma es la adecuación funcional, que evalúa la facultad del software para brindar soluciones que atienden las necesidades especificadas.

Esto incluye subcaracterísticas como la completitud funcional, que asegura que todas las tareas necesarias están cubiertas, la corrección funcional, que verifica la precisión de los resultados, y la adecuación funcional, que mide la efectividad de las funciones para lograr objetivos específicos (Britton, 2021).

Otra característica crucial es la eficiencia del desempeño, que se centra en cómo el software utiliza los recursos en relación con su rendimiento. Esto incluye la conducta temporal, el uso de recursos y la capacidad del sistema. La eficiencia del rendimiento es clave para asegurar que el software logre manejar grandes volúmenes de datos y realizar tareas complejas sin comprometer la velocidad ni la estabilidad (Rebeś, 2023).

La usabilidad es otra característica importante, que evalúa la capacidad de los usuarios para dominar rápidamente el uso del software y su efectividad para cumplir sus objetivos. Subcaracterísticas como reconocibilidad de la adecuación, la aprendibilidad y la operabilidad aseguran que el software sea intuitivo, fácil de aprender y operar, y que minimice los errores del usuario, lo cual es crucial para la satisfacción del usuario final (ASQ, 2023).

Para validar los resultados de la calidad de uso, se puede aplicar la característica de satisfacción utilizando la subcaracterística de utilidad según la norma. La satisfacción mide cómo el software cumple con las demandas de los usuarios en cuanto a eficiencia, efectividad y satisfacción general. La subcaracterística de utilidad evalúa si el software proporciona beneficios y cumple con las expectativas de los usuarios en relación con su situación de uso, asegurando que los resultados sean útiles y relevantes para los objetivos específicos del usuario (Britton, 2021).

Finalmente, la seguridad del software es esencial para asegurar la protección de la información y los datos contra accesos ilegales. Las Subcaracterísticas de seguridad incluyen la confidencialidad, que asegura que los datos solo sean accesibles para personas autorizadas, la integridad, que previene el acceso no permitido y la modificación de información, y la autenticidad, que autentica la identidad de los usuarios y recursos. Estas

medidas son cruciales para mantener la confianza en el software y asegurar que los datos sensibles estén protegidos (Rebeés, 2023).

1.4.3. Evaluación de la calidad de uso según la norma ISO/IEC:25010.

Con el fin de medir la calidad de la aplicación móvil en el contexto de la tesis, se aplicará los principios y métricas de la ISO, enfocándose en la calidad de uso y la satisfacción del usuario a través de la Subcaracterística de utilidad (Ver **Figura 12**).

Figura 12

Calidad de Uso 25010

ISO 25010 - Calidad de uso



Nota. Adaptado de (Paneiva, 2023).

La adecuación funcional es fundamental para asegurar que la aplicación y el administrador web cumplen con los requisitos específicos de los usuarios. Un estudio sobre aplicaciones de salud reproductiva muestra que el modelo ISO/IEC 25010 permite identificar y evaluar características clave como la exactitud y la consistencia de los datos. Este enfoque

asegura que funcionalidades de FR y AR en la aplicación móvil y el administrador web sean precisas y efectivas (Moumane et al., 2024).

El grado de satisfacción del usuario, medida a través de la subcaracterística de utilidad, como se indica en la **Figura 13**, es fundamental para validar la calidad en uso de la app. La revisión de aplicaciones móviles con relación de videos demuestra que esta métrica es esencial para entender cómo las funcionalidades del software cumplen con las expectativas de los usuarios. Aplicar esta evaluación permitirá asegurar que las soluciones desarrolladas no solo son técnicamente correctas, sino también útiles y satisfactorias para los usuarios dentro del campus universitario (Hussain et al., 2018).

Figura 13

Satisfacción y utilidad

Satisfacción	Utilidad	Nivel de satisfacción	¿Qué tan satisfecho está el usuario?	Realizar un cuestionario sobre el nivel de satisfacción sobre el sistema.	Uso	$X = A/B$ A= Número de preguntas con respuesta satisfactorias B = Número total de preguntas realizadas en el cuestionario. Dónde: $B > 0$
		Uso discrecional de las funciones	¿Qué porcentaje de los usuarios optan por utilizar las funciones sistema?	Observación de uso	Uso	$X = A/B$ A= Número de funciones específicas del software que se utilizan B= Número total de funciones que están destinados a ser usados Dónde: $B > 0$
		Porcentaje de quejas de los clientes	¿Cuál es el porcentaje de quejas realizadas por los clientes?	Contar el número de clientes que se quejan y contar el número total de clientes	Uso	$X = A/B$ A = Número de clientes que se quejan B = Número total de clientes Dónde: $B > 0$

Nota. (Vaca, 2017).

CAPÍTULO 2

Desarrollo del proyecto

2.1 Planificación

2.1.1 Historias de usuario

Son una herramienta fundamental para SCRUM, sirve para la captura de requisitos. Dichas historias facilitan que los desarrolladores logren comprender y documentar las necesidades y expectativas de los usuarios de manera clara y concisa (Cohn & Beck, 2008).

Según Suaza (2013), las historias de usuario facilitan la interacción entre los interesados y los desarrolladores, promoviendo un avance incremental del software con iteraciones cortas. Además, estas historias ayudan a reducir la ambigüedad en la especificación de requisitos, dando como resultado un proyecto ágil exitoso.

A continuación, la **Tabla 2**, detalla la historia de usuario número 1, con los siguientes puntos a considerar: ID, nombre, descripción y validación.

Tabla 2

Historia de usuario 01 – Subir imágenes de estudiantes al administrador web

Historia de Usuario	
ID: HU-01	Usuario: Administrador del sistema
Nombre de historia: Subir imágenes de estudiantes al administrador web	
Descripción: En calidad de administrador del sistema, necesito subir una imagen de una persona junto con su información personal (nombre, ID, facultad, etc.) al administrador web, con la finalidad de que esa imagen junto con su información sea registrada en la base de datos y luego utilizada para el FR y AR.	
Validación:	

- Puedo seleccionar una imagen desde mi equipo.
- Puedo ingresar la información de la persona asociada a la imagen.
- Recibo una confirmación de que la imagen y la información se han subido correctamente.

Nota. Autoría propia.

A continuación, la **Tabla 3**, detalla la historia de usuario número 2, con los siguientes puntos a considerar: ID, nombre, descripción y validación.

Tabla 3

Historia de usuario 02 – Recargar los modelos de reconocimiento facial

Historia de Usuario

ID: HU-02

Usuario: Administrador del sistema

Nombre de historia: Recargar los modelos de reconocimiento facial

Descripción:

En calidad de administrador del sistema, quiero recargar los modelos de FR cada vez que se suban nuevas imágenes al sistema, con la finalidad de que la API actualice la base de datos de rostros y pueda identificar nuevos registros.

Validación:

- Puedo enviar una petición a la API para recargar los modelos.
- La API debe actualizarse correctamente con las nuevas imágenes sin afectar la funcionalidad en curso.
- Recibo una confirmación de que los modelos se han recargado exitosamente.

Nota. Autoría propia.

A continuación, la **Tabla 4**, detalla la historia de usuario número 3, con los siguientes puntos a considerar: ID, nombre, descripción y validación.

Tabla 4

Historia de usuario 03 – Visualizar y administrar imágenes junto sus datos

Historia de Usuario	
ID: HU-03	Usuario: Administrador del sistema
Nombre de historia: Visualizar y administrar imágenes junto sus datos	
Descripción: En calidad de administrador del sistema, necesito eliminar la imagen y los datos personales de una persona del sistema, con la finalidad de garantizar que la base de datos se mantenga al día, libre de información obsoleta.	
Validación: <ul style="list-style-type: none">• Puedo seleccionar una imagen y los datos asociados para eliminarlos.• Recibo una confirmación de que la imagen y la información han sido eliminadas correctamente.	

Nota. Autoría propia.

A continuación, la **Tabla 5**, detalla la historia de usuario número 4, con los siguientes puntos a considerar: ID, nombre, descripción y validación.

Tabla 5

Historia de usuario 04 - Identificar una persona en tiempo real

Historia de Usuario	
ID: HU-04	Usuario: Administrador del sistema
Nombre de historia: Identificar una persona en tiempo real	

Descripción:

En calidad de administrador del sistema, quiero procesar imágenes capturadas en tiempo real desde un video (separando cada rostro en una imagen), con la finalidad de comparar esos rostros con los almacenados en la nube y devolver los datos respectivos de la persona identificada.

Validación:

- La API debe recibir la imagen desde el video, procesarla y buscar coincidencias.
- Si el rostro es identificado, la API debe devolver los datos de la persona asociada.
- La respuesta debe ser rápida para asegurar que el proceso se realiza en tiempo real.

Nota. Autoría propia.

A continuación, la **Tabla 6**, detalla la historia de usuario número 5, con los siguientes puntos a considerar: ID, nombre, descripción y validación.

Tabla 6

Historia de usuario 05 - Identificar una persona y mostrar su información en realidad aumentada

Historia de Usuario

ID: HU-05**Usuario:** Usuario**Nombre de historia:** Identificar una persona y mostrar su información en realidad aumentada**Descripción:**

Como usuario, quiero escanear el rostro de una persona usando la cámara de mi dispositivo móvil, que la aplicación identifique a la persona en tiempo real mediante el API de FR, y que la información registrada de la persona se muestre en realidad aumentada, con la

finalidad de visualizar los datos personales de la persona identificada en el campus de la universidad de manera interactiva y eficiente.

Validación:

- La aplicación debe acceder a la cámara del dispositivo móvil para capturar el rostro en tiempo real.
 - La imagen capturada debe ser enviada a la API de FR.
 - La API debe procesar la imagen y buscar coincidencias en la base de datos de imágenes previamente registradas.
 - Si la persona es identificada, la aplicación debe recibir y mostrar la información de esa persona en la pantalla mediante AR (nombre, facultad, ID, etc.).
 - Si no se identifica a la persona, se debe mostrar un mensaje que indique que no está registrada en el sistema.
 - Toda la comunicación entre la aplicación móvil y la API debe realizarse de manera transparente y eficiente.
-

Nota. Autoría propia.

2.1.2 Estimación de esfuerzo

En Scrum, la estimación de esfuerzo con “t-shirt sizing” es una técnica intuitiva y simple que se utiliza para evaluar el esfuerzo relativo necesario para completar los elementos del Product Backlog. En lugar de usar números, se asignan tamaños de camiseta como “Extra-Small (XS)”, “Small (S)”, “Medium (M)”, “Large (L)”, “Extra Large (XL)” y “Double Extra Large (XXL)” a cada tarea. Esta técnica permite pensar de manera más dinámica sobre el esfuerzo requerido, considerando factores como la complejidad, el tiempo y los recursos necesarios. Al eliminar la puntuación numérica, los miembros del equipo pueden visualizar mejor la magnitud del trabajo y comunicarse de manera más efectiva sobre las prioridades y la carga de trabajo (Mahnič & Hovelja, 2012).

En la **Tabla 7** se explica de manera más visual cómo funciona esta valoración del esfuerzo.

Tabla 7

Estimación de esfuerzo t-shirt sizing

Valor	Estimación
XS	1 – 2 horas
S	2 - 4 horas
M	4 - 8 horas
L	8 - 16 horas
XL	16 – 24 horas
XXL	24 – 40 horas

Nota. Adaptado de (Talreja, 2023).

2.1.3 Prioridad de historias de usuario

En Scrum, la prioridad de las historias de usuario es fundamental para asegurar que se trabaje en las tareas más importantes y valiosas primero. La priorización se basa en varios factores, como el valor de negocio, la urgencia y la complejidad técnica. La técnica que comúnmente se utiliza es el método MoSCoW (Ver **Tabla 8**), que clasifica las historias en categorías de “Must have” (debe tener), “Should have” (debería tener), “Could have” (podría tener) y “Won’t have” (no tendrá). Este enfoque ayuda a enfocarse en las funcionalidades esenciales que aportan el mayor valor al producto y a los usuarios finales (Racheva et al., 2010).

Tabla 8*Estimación de prioridad MoSCoW*

Prioridad	Descripción
Must have	Funcionalidades esenciales que son críticas para el éxito del proyecto. Sin ellas, el producto no puede ser lanzado. Ejemplo: Autenticación de usuario.
Should have	Funcionalidades importantes, pero no críticas. Aportan un valor significativo, pero el producto puede funcionar sin ellas en una primera versión. Ejemplo: Notificaciones por correo electrónico.
Could have	Funcionalidades deseables que mejoran la experiencia del usuario, pero no son esenciales. Ejemplo: Personalización del perfil de usuario.
Won't have	Funcionalidades que no se incluirán en esta versión del producto, pero podrían considerarse para futuras versiones. Ejemplo: Integración con redes sociales.

Nota. Adaptado de (Racheva et al., 2010).

2.1.4 Creación del Product Backlog

El Product Backlog es una lista priorizada y emergente de todo el trabajo que podría ser necesario para mejorar un producto. Esta lista incluye nuevas características, mejoras, correcciones de errores y tareas técnicas. El Product Owner es responsable de mantener y ordenar el backlog, asegurando que los elementos más valiosos y urgentes estén en la parte superior. La priorización y refinamiento continuo del Product Backlog permiten al equipo de desarrollo enfocarse en entregar el máximo valor en cada sprint (Schwaber & Sutherland, 2020).

En la **Tabla 9** se detalla el Product Backlog diseñada para este proyecto.

Tabla 9

Product Backlog

ID	Alias	Prioridad	Esfuerzo
HU-01	Subir imágenes de estudiantes al administrador web	Must have	L
HU-02	Recargar los modelos de reconocimiento facial	Must have	S
HU-03	Visualizar y administrar imágenes junto sus datos	Must have	XL
HU-04	Identificar una persona en tiempo real	Must have	L
HU-05	Identificar una persona y mostrar su información en realidad aumentada	Must have	XL

Nota. Autoría propia.

2.1.5 Definición de roles del proyecto

La **Tabla 10** ,presenta la definición de roles dentro del proyecto, especificando las responsabilidades y funciones de cada miembro del equipo. Esta estructura es fundamental para asegurar una clara distribución de tareas y una gestión eficiente durante el desarrollo del proyecto, basado en SCRUM.

Tabla 10

Roles del proyecto

Rol	Miembro	Descripción
Product Owner	Sr. Roberth Gabriel Lima	Responsable del desarrollo del sistema de reconocimiento facial y realidad aumentada.
Scrum Master	MSc. Fausto Salazar	Director del trabajo de titulación

Nota. Autoría propia.

2.1.6 Planificación de Sprints

La **Tabla 11** detalla la planificación de los Sprints, especificando las historias de usuario correspondiente a cada sprint, junto a su lapso estimado para realizarlo. Esta planificación es crucial para garantizar un desarrollo organizado y enfocado en la entrega de valor de manera iterativa durante el ciclo del proyecto.

Tabla 11

Planificación de Sprints

ID	Alias	Sprint	Fecha
HU-01	Subir imágenes de estudiantes al administrador web	0	01/07/2024 - 31/07/2024
HU-02	Recargar los modelos de reconocimiento facial		
HU-03	Visualizar y administrar imágenes junto sus datos		
HU-04	Identificar una persona en tiempo real	1	01/08/2024 - 31/08/2024
HU-05	Identificar una persona y mostrar su información en realidad aumentada	2	01/09/2024 - 11/11/2024

Nota. Autoría propia.

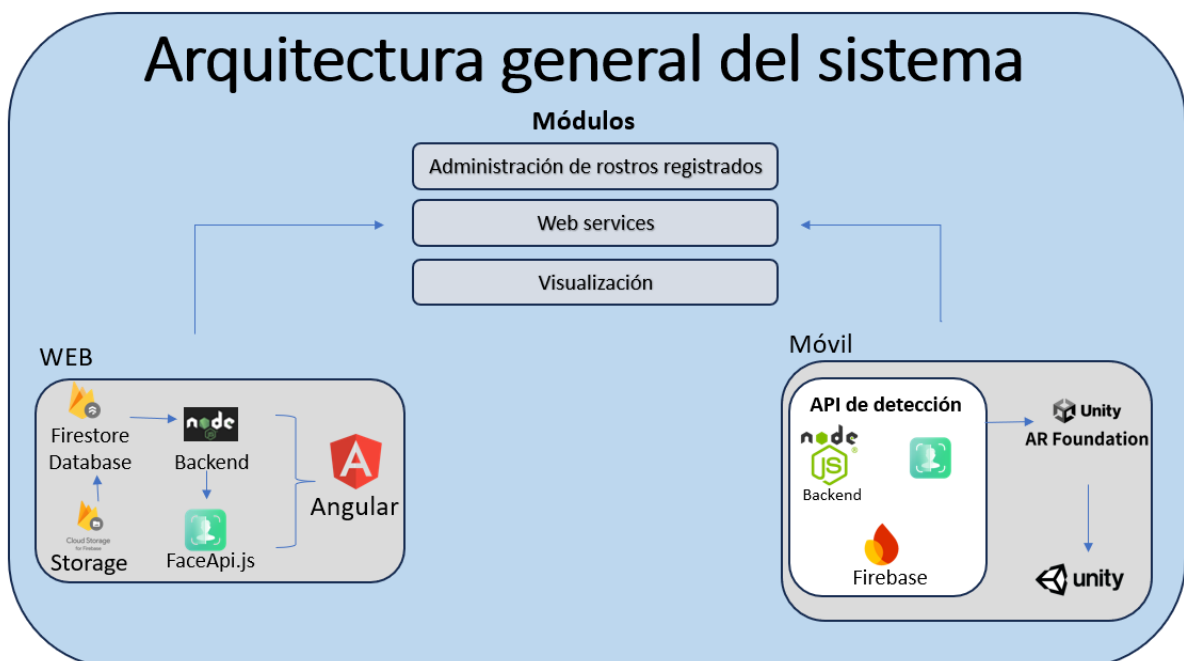
2.2. Diseño

2.2.1. Arquitectura del sistema

El diseño de este proyecto utiliza una arquitectura cliente-servidor que se divide en tres módulos principales: Administrador Web, Web services (API REST) y Visualización. (Ver **Figura 14**). La arquitectura se organiza en dos partes fundamentales: el componente móvil y el componente API REST, los cuales interactúan a través de un sistema de servicios que facilitan el reconocimiento y la visualización de la información facial.

Figura 14

Arquitectura del sistema



Nota. Autoría propia.

En el lado web, encontramos el administrador web, que está construido con Angular en el frontend y un backend desarrollado en Node.js. La administración de los rostros registrados es una tarea clave de este sistema, gestionando la información guardada en Firestore Database y el almacenamiento de imágenes en Cloud Storage.

El backend utiliza la librería face-api.js para la detección y el FR, realizando la identificación de personas de manera individual.

Por otro lado, la aplicación móvil, desarrollada en Unity y utilizando AR Foundation, consume el API REST implementado en Node.js. Esta API, además de integrar la face-api.js, se comunica con Firebase para obtener y almacenar los datos. El dispositivo móvil captura imágenes en tiempo real, realiza la detección y FR, y utiliza AR Foundation para mostrar la información de la persona identificada directamente en la interfaz de la aplicación.

Esta arquitectura distribuida permite que el administrador web y la aplicación móvil compartan un núcleo común a través de los servicios web, asegurando que los datos de FR estén sincronizados y disponibles en ambas plataformas de manera eficiente y en tiempo real.

2.2.2. Actores

Seguidamente, se presentan los diagramas de secuencia (**Figura 20 - Figura 24**) que ilustran los flujos de interacción entre los componentes del sistema. Dentro del mismo vamos a tener algunos actores, que se presentan detalladamente en la **Tabla 12**.

Tabla 12

Actores

Actor	Actividades
Administrador	Cargar imágenes y registrar datos de personas asociadas.
	Recargar los modelos de los rostros registrados
	Visualizar, administrar y eliminar información de personas registradas.
	Identificación en tiempo real y despliegue de datos mediante la cámara.
Usuario	Entrar a la aplicación móvil, iniciar el proceso de identificación y visualizar información mediante realidad aumentada.

Nota. Autoría propia.

2.2.3. Diagramas de casos de uso

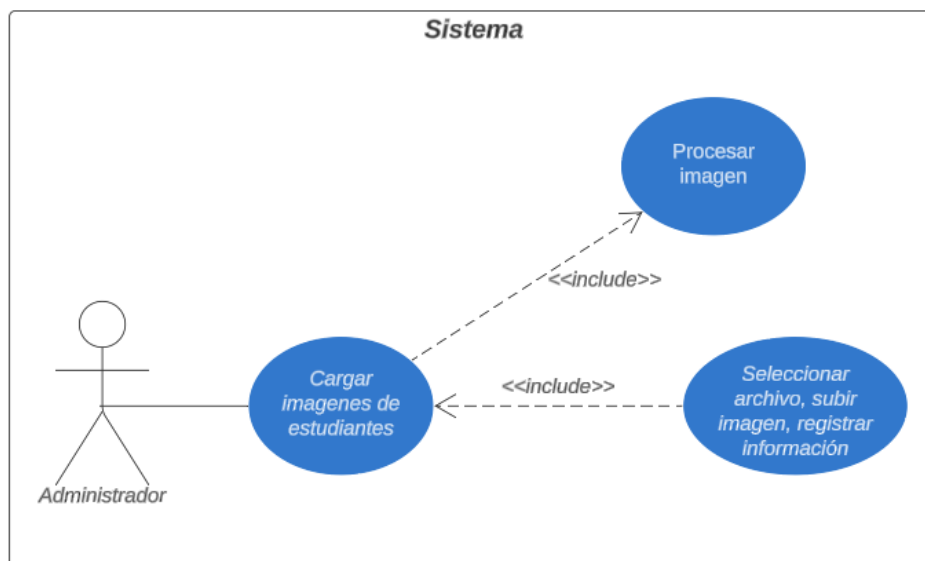
En este apartado se muestran los casos de uso relevantes con respecto a todas las funcionalidades del sistema y la aplicación móvil (que consume el API).

Para esto, se utilizan diagramas UML que ilustran las distintas actividades llevadas a cabo por los actores que se presentaron en la **Tabla 12** y poder tener un mejor entendimiento de cómo funciona y se conecta todo el sistema en sí.

En la **Figura 15**, se muestra al usuario tratando de subir una nueva imagen junto con sus datos, al administrador web, este diagrama de casos de uso solo es realizado por el administrador encargado.

Figura 15

Diagrama de caso de uso 1, Cargar imágenes y registrar datos de personas asociadas.



Nota. Autoría propia.

En la **Tabla 13** se detalla la información del primer caso de uso, cargar imágenes de estudiantes

Tabla 13

Caso de uso número 1, Cargar imágenes y registrar datos de personas asociadas.

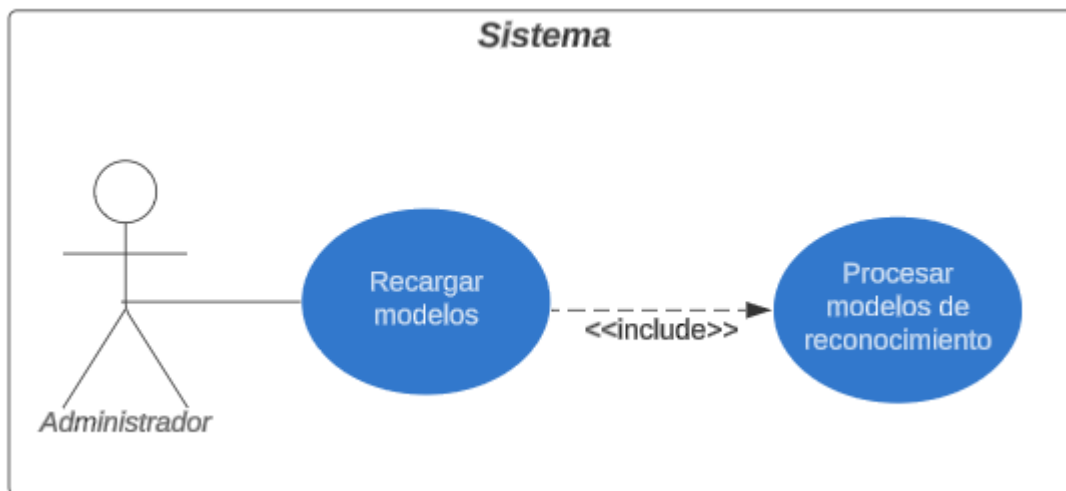
Caso de uso número 1	
Nombre:	Cargar imágenes y registrar datos de personas asociadas.
Descripción:	Permite al administrador seleccionar y subir imágenes de estudiantes al sistema junto con sus datos.
Actor:	Administrador
Precondiciones:	El sistema está en línea y listo para procesar las imágenes.
Flujo normal:	<ol style="list-style-type: none">1. El usuario debe seleccionar una imagen local previamente capturada2. El sistema procesa la imagen3. El sistema muestra un formulario de datos importantes4. El administrador llena los campos5. El sistema muestra una ventana de éxito
Flujo alternativo	2.A. Si el sistema encuentra un error durante el procesamiento de la imagen, muestra un mensaje de error y permite al administrador intentarlo nuevamente
Postcondiciones	La imagen del estudiante y la información asociada quedan almacenadas correctamente en la base de datos El administrador puede visualizar la imagen y los datos registrados en la interfaz de administración

Nota. Autoría propia.

En la **Figura 16**, se observa al usuario tratando de recargar los modelos del API, dentro del administrador web, este diagrama de casos de uso solo es realizado por el administrador encargado.

Figura 16

Diagrama de caso de uso 2, recargar modelos de reconocimiento facial



Nota. Autoría propia.

En la **Tabla 14**, se describe la información del segundo caso de uso denominado recargar modelos de FR.

Tabla 14

Caso de uso número 2, recargar modelos de reconocimiento facial

Caso de uso número 2	
Nombre:	Recargar modelos de reconocimiento facial
Descripción:	El administrador puede recargar los modelos de FR en el sistema para garantizar que están actualizados y listos para usar
Actor:	Administrador
Precondiciones:	El sistema debe estar operativo

Se deben de haber agregado nuevos rostros (de preferencia varios)

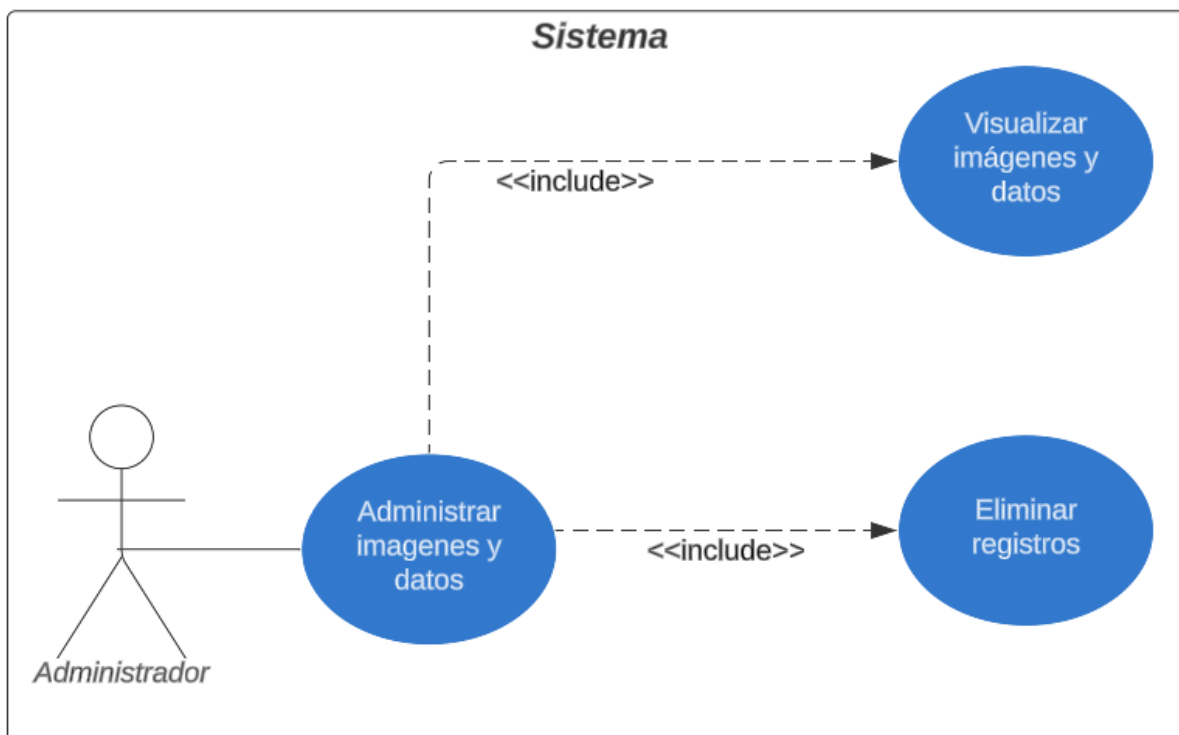
- Flujo normal:**
1. El administrador selecciona la opción para recargar los modelos
 2. El sistema recarga los modelos desde el conjunto de datos almacenado
 3. El sistema muestra un formulario de datos importantes
 4. El sistema confirma que los modelos han sido recargados correctamente
- Flujo alternativo**
- 2.A. Si el sistema encuentra un error durante la recarga de los nuevos modelos, muestra un mensaje de error y permite al administrador intentarlo nuevamente
- Postcondiciones** Los modelos de FR se actualizan y están listos para ser usados

Nota. Autoría propia.

En la **Figura 17**, se observa al usuario tratando gestionar los registros que contienen las imágenes y datos correspondientes a las mismas, teniendo la opción de eliminar o visualizar, dentro del administrador web, este diagrama de casos de uso solo es realizado por el administrador encargado.

Figura 17

Diagrama de caso de uso 3, administrar imágenes y datos



Nota. Autoría propia.

En la **Tabla 15**, se describe la información del tercer caso de uso denominado administrar imágenes y datos.

Tabla 15

Caso de uso número 3, *administrar imágenes y datos*

Caso de uso número 3	
Nombre:	Administrar imágenes y datos
Descripción:	El administrador puede visualizar y eliminar las imágenes junto con la información asociada a los estudiantes en el sistema
Actor:	Administrador
Precondiciones:	Deben existir imágenes y datos almacenados en el sistema
Flujo normal:	1. El sistema muestra una tabla de imágenes con su información asociada

2. El administrador puede seleccionar una fila para eliminar
3. El sistema confirma la acción y realiza la operación correspondiente

Flujo alternativo 3.A. Si el sistema encuentra un error durante la operación, muestra un mensaje de error

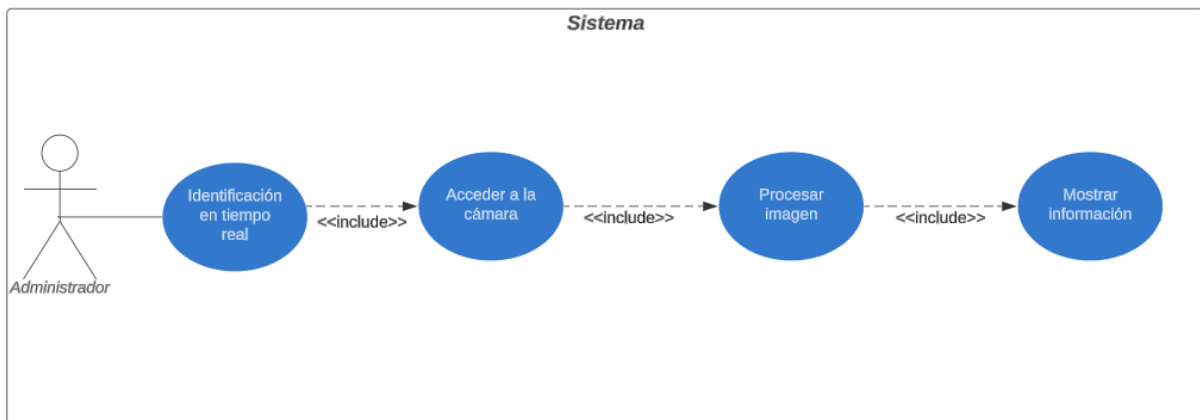
Postcondiciones Los registros seleccionados han sido eliminados

Nota. Autoría propia.

En la **Figura 18**, se observa al usuario tratando identificar a una persona en tiempo real con la cámara de la computadora, dentro del administrador web, para corroborar que el sistema está conectado perfectamente a la base de datos, este diagrama de casos de uso solo es realizado por el administrador encargado.

Figura 18

Diagrama de caso de uso 4, identificación en tiempo real



Nota. Autoría propia.

En la **Tabla 16**, se describe la información del cuarto caso de uso denominado identificación en tiempo real.

Tabla 16

Caso de uso número 4, identificación en tiempo real

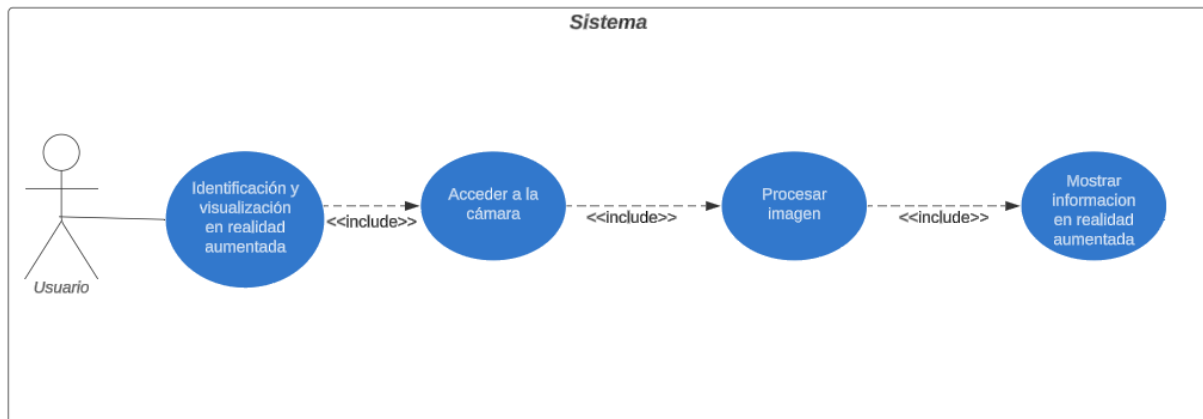
Caso de uso número 4	
Nombre:	Identificación en tiempo real
Descripción:	El administrador puede identificar a una persona utilizando la cámara en tiempo real, y el sistema muestra la información correspondiente una vez que se ha hecho la identificación
Actor:	Administrador
Precondiciones:	La cámara debe estar disponible y el sistema debe tener acceso a la base de datos de imágenes y datos
Flujo normal:	<ol style="list-style-type: none">1. El administrador selecciona la opción para iniciar la identificación2. El sistema muestra la transmisión en tiempo real de la cámara3. El administrador pulsa el botón de "Identificar"4. El sistema procesa la imagen y busca una coincidencia en la base de datos5. Si se encuentra una coincidencia, el sistema muestra la información de la persona identificada
Flujo alternativo	No hay
Postcondiciones	La información de la persona identificada aparece en pantalla

Nota. Autoría propia.

En la **Figura 19**, se observa al usuario final de la aplicación móvil, tratando de realizar el reconocimiento e identificación facial, esperando como respuesta la información en tiempo real mostrada con AR.

Figura 19

Diagrama de caso de uso 5, identificación y visualización en realidad aumentada



Nota. Autoría propia.

En la **Tabla 17**, se describe la información del primer caso de uso denominado cargar imágenes de estudiantes

Tabla 17

Caso de uso número 5, identificación y visualización en realidad aumentada

Caso de uso número 5	
Nombre:	Identificación y visualización en realidad aumentada
Descripción:	El usuario abre la aplicación móvil, coloca su rostro frente a la cámara, y si el sistema lo identifica muestra su información en tiempo real en un entorno de AR.
Actor:	Usuario
Precondiciones:	La cámara debe estar disponible, y el sistema debe tener acceso a la base de datos de imágenes y datos El API debe estar en línea
Flujo normal:	<ol style="list-style-type: none">1. El usuario abre la aplicación2. El sistema solicita permiso para acceder a la cámara

3. El sistema muestra la transmisión en tiempo real de la cámara
4. El usuario pulsa el botón de "Identificar"
5. El sistema procesa la imagen y busca una coincidencia en la base de datos
6. Si se encuentra una coincidencia, el sistema muestra la información en la pantalla mediante AR

Flujo alternativo 5.A. Demora en la respuesta de la API, el usuario deberá pulsar de nuevo el botón de identificar

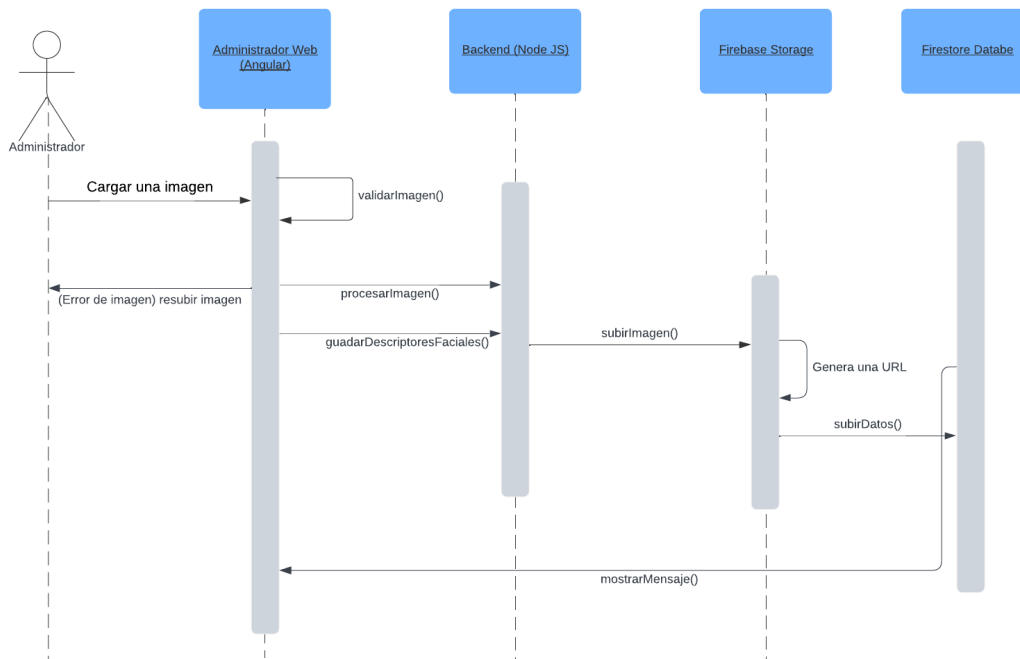
Postcondiciones La información de la persona identificada aparece superpuesta en la vista de AR

Nota. Autoría propia.

2.2.4. Diagramas de Secuencia

Figura 20

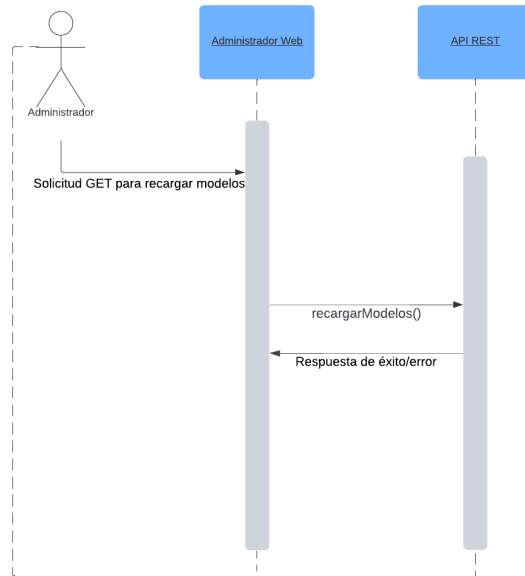
Diagrama de secuencia – Cargar imágenes de estudiantes con su información



Nota. Autoría propia.

Figura 21

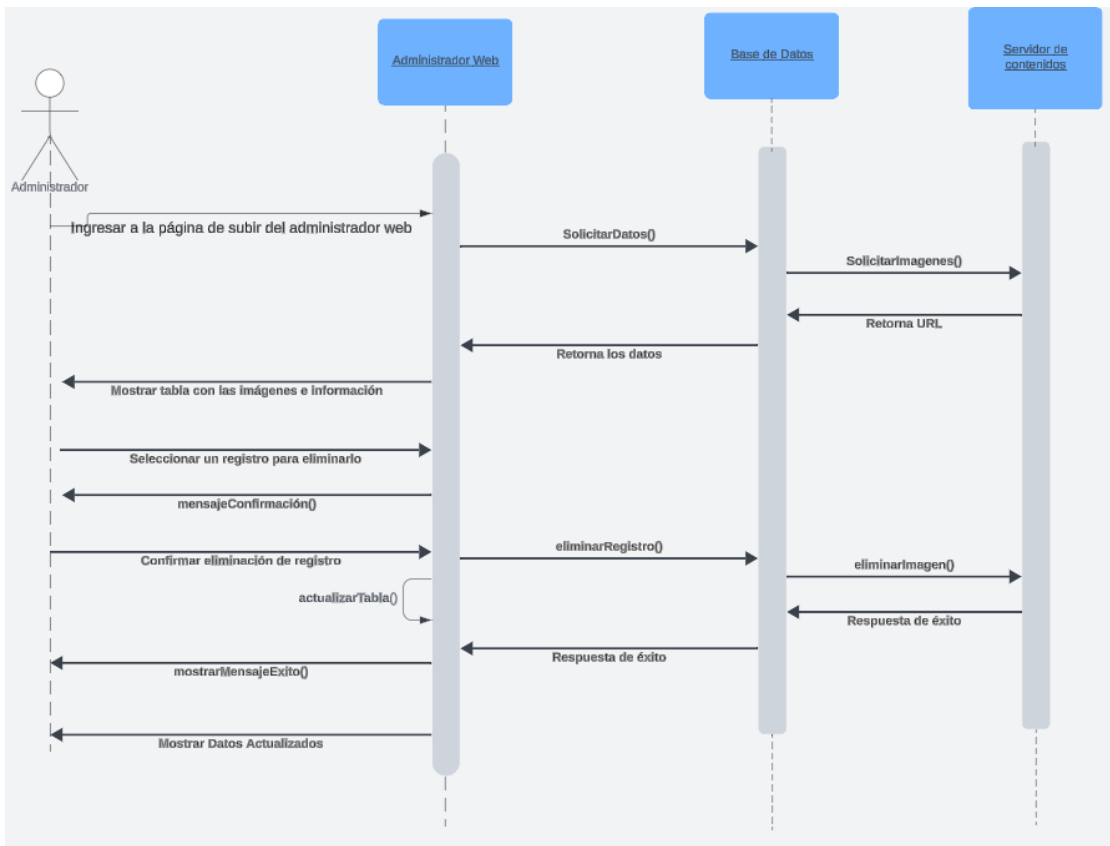
Diagrama de secuencia – Recargar modelos de reconocimiento facial



Nota. Autoría propia.

Figura 22

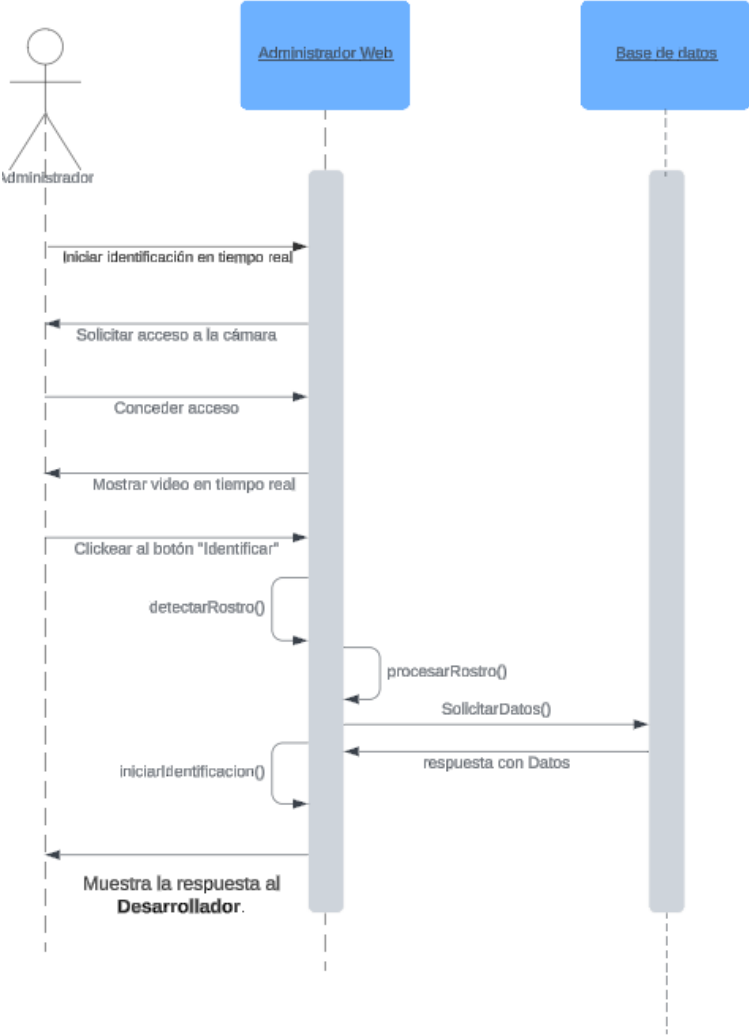
Diagrama de secuencia – Administrar imágenes y datos



Nota. Autoría propia.

Figura 23

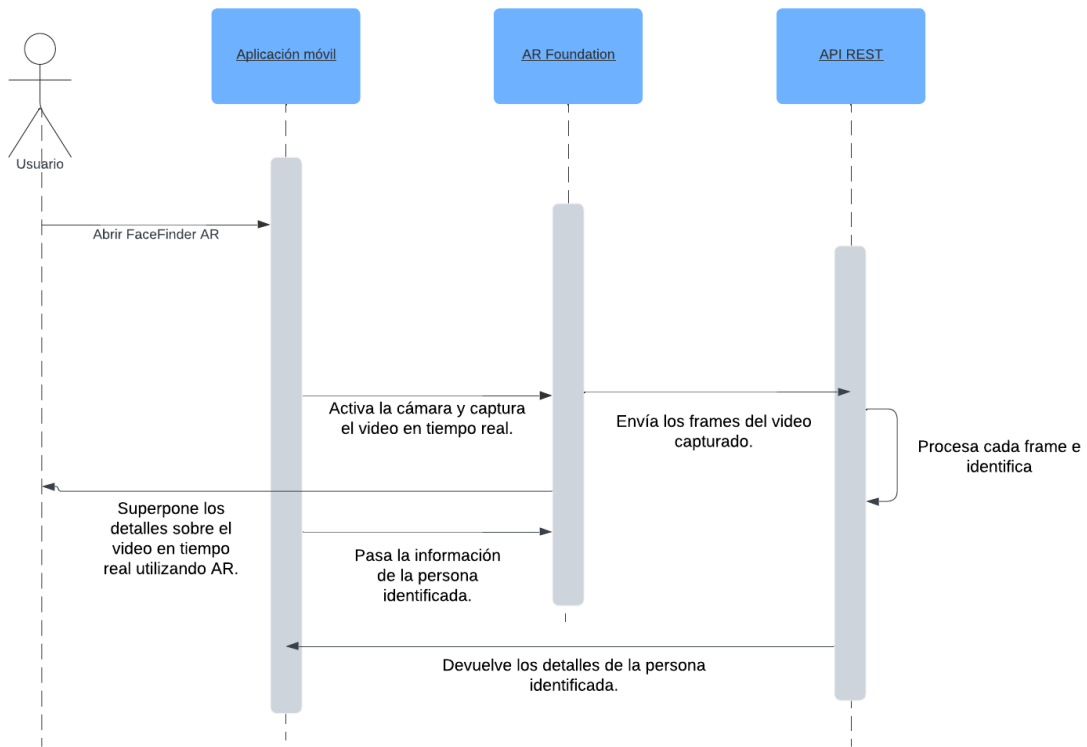
Diagrama de secuencia – Identificación en tiempo real



Nota. Autoría propia.

Figura 24

Diagrama de secuencia – Identificación y visualización en realidad aumentada



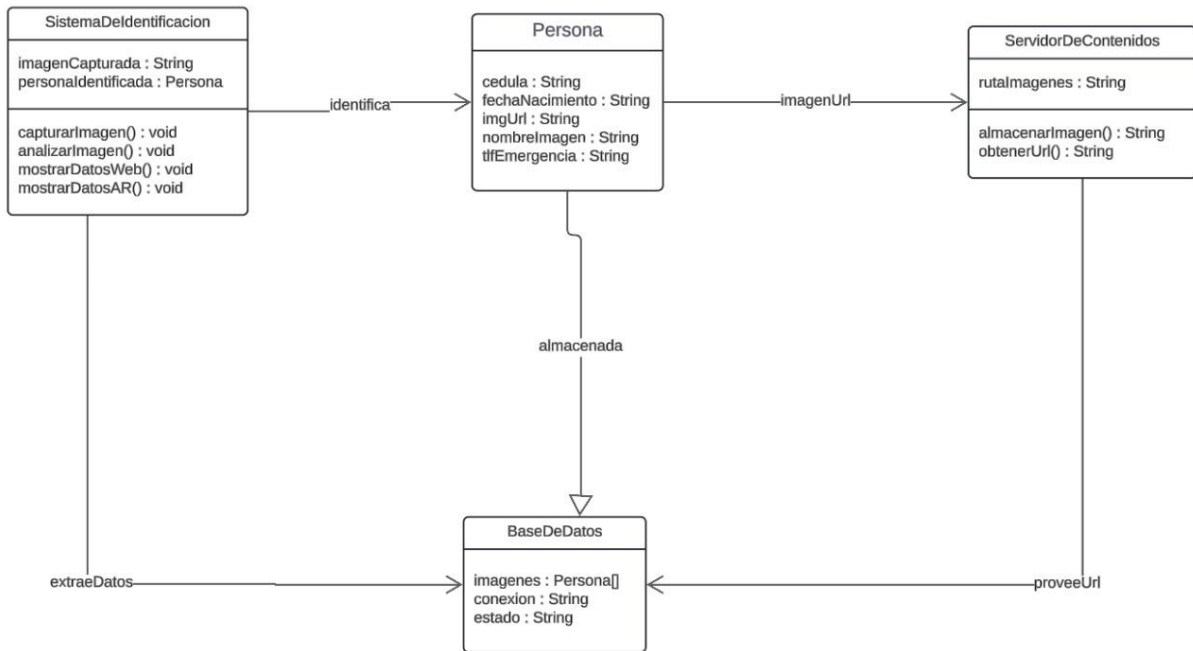
Nota. Autoría propia.

2.2.5. Diagrama de clases

El diagrama de clases proporciona una representación detallada de los componentes que conformarán el sistema y sus funciones principales. Esto nos permite tener una comprensión clara de la estructura de cada clase dentro del proyecto de desarrollo, como se ilustra en la **Figura 25**.

Figura 25

Diagrama de clases



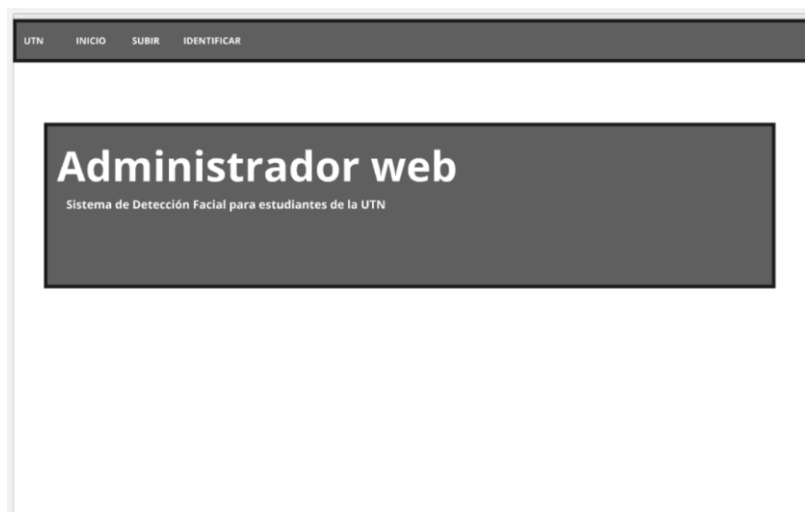
Nota. Autoría propia.

2.2.6. Prototipo del Administrador web

En la **Figura 26**, se presenta el prototipo del administrador web, la página de Inicio, el cual ilustra la página inicial que se plantea realizar

Figura 26

Prototipo Administrador web - Inicio



Nota. Autoría propia.

En la **Figura 27**, se presenta el prototipo del administrador web, la página de Subir, el cual ilustra la página en la cual se realiza las actividades de subir imágenes, eliminar imágenes, y recargar los modelos, esto es con la finalidad de que sea bastante amigable con la persona encargada de administrar dicho sistema.

Figura 27

Prototipo Administrador web - Subir



Nota. Autoría propia.

En la **Figura 28**, se presenta el prototipo del administrador web, la página de Identificar, el cual ilustra la página encargada de comprobar las funciones previas a esta página.

Figura 28

Prototipo Administrador web - Identificar



Nota. Autoría propia.

2.2.7. Prototipo de la aplicación móvil

En la **Figura 29**, se presenta el prototipo de la aplicación móvil, el cual es pensado para que sea lo más simple posible, y funcional a la hora de realizar el FR/identificación facial.

Figura 29

Prototipo Aplicación móvil – Screen 1



Nota. Autoría propia.

En la **Figura 30**, se presenta el prototipo de la aplicación móvil, el cual es pensado para que sea lo más simple posible, y funcional a la hora de realizar el FR/identificación facial, mostrando como se verían los datos con AR.

Figura 30

Prototipo aplicación móvil – Screenshot 2



Nota. Autoría propia.

2.3. Desarrollo del sistema

2.3.1. Módulos del proyecto

Para el correcto funcionamiento de todo el proyecto, se compone de varios módulos. A continuación, en la **Tabla 18** se describen los módulos que estarán presentes.

Tabla 18*Módulos del proyecto*

Módulo	Descripción	Funcionalidades
Administrador Web	Interfaz para gestionar usuarios y datos del sistema.	<ul style="list-style-type: none"> • Subir imágenes • Eliminar imágenes • Detección facial
Web services (API)	Interfaz de programación que permite la comunicación entre el frontend y backend.	<ul style="list-style-type: none"> • Gestión de usuarios • Identificación facial • Actualización de datos
Aplicación Móvil	Plataforma móvil que utiliza FR y AR	<ul style="list-style-type: none"> • Visualización de datos • Interacción en AR

Nota. Autoría propia.**2.3.2. Sprint 0**

A continuación, en la **Tabla 19**, se presenta la tabla del Sprint Backlog del Sprint 0, que detalla las historias de usuario y las tareas asignadas para el presente sprint, así como las estimaciones de tiempo correspondientes. Este documento es esencial para la gestión y seguimiento del avance del proyecto.

Tabla 19*Sprint 0*

SPRINT BACKLOG				
Sprint: 0				
Total horas: 150				
Fecha Inicio SP0: 01/07/2024				
Fecha Final SP0: 31/07/2024				
Actividad / Historia de usuario	Fase Desarrollo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO REAL (Horas)
Creación del proyecto	Desarrollo	Creación y configuración del proyecto	2	0.5
	Análisis	Analizar Firebase (Firestore y Storage)	5	4
HU-01 Subir imágenes de estudiantes al administrador web	Análisis	Revisar cómo funcionan las conexiones y su relación	5	5
	Desarrollo	Diseño del componente de carga de imágenes	8	9
	Desarrollo	Implementación de la lógica de carga	10	11

	Desarrollo	Integración de FaceAPI.js para procesamiento	15	12
	Desarrollo	Pruebas y ajustes	8	3
HU-02 Recargar los modelos de reconocimiento facial	Análisis	Conoces cual es la configuración necesaria a futuro para la utilización de la App	5	4
	Desarrollo	Diseño del botón en el administrador web	3	2
	Desarrollo	Implementación de la lógica de llamada a la API	14	15
	Desarrollo	Creación de la API en Node.js	24	25
	Desarrollo	Integración de la respuesta de la API	10	2
	Desarrollo	Pruebas y ajustes	6	01
	HU-03 Visualizar y administrar imágenes junto sus datos	Desarrollo	Diseño de la tabla para visualizar registros	4
Desarrollo		Implementación de la lógica para recuperar datos de la base de datos	15	10
Desarrollo		Integración de los datos en la tabla	5	11
Desarrollo		Implementación de la funcionalidad de eliminación de registros	5	5
Desarrollo		Pruebas y ajustes	6	5
Reuniones	Gestión	Planificación	0.1	0.1
	Gestión	Diarias	0.1	0.1
	Gestión	Revisión	0.1	0.1
	Gestión	Retrospectiva	0.1	0.1
TOTAL			150	129

Nota. Autoría propia.

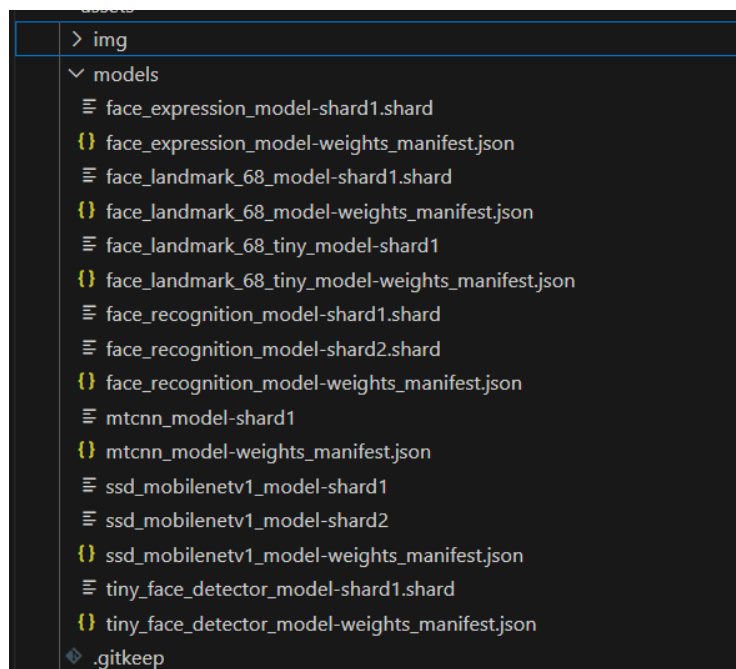
Para el Sprint 0 como se presenta en la **Tabla 19**, primero se tuvo que crear un proyecto, usando Visual Studio Code, en su última versión, dentro de este se iban a descargar todas las herramientas necesarias (Angular, Node JS, Firebase y fase-api.js) para poder empezar a codificar las funciones que tendrá el administrador web: subir imágenes junto con información de la persona dentro de la imagen, poder borrar en conjunto la información y la imagen, de igual forma un botón que recarga los modelos con los subidos recientemente por

el administrador, para que se puedan utilizar por la aplicación móvil, por ultimo programar una pestaña en la que se podrá poder verificar tanto el consumo del fase-api.js como la conexión y detección de rostros al momento de comparar con la base de datos y el video en tiempo real, llamada identificar.

Para evitar problemas de congelamientos del api fase-api.js, la cual se está consumiendo, se descargaron todos los modelos que permiten el FR (no hay que confundir reconocimiento facial con identificación facial) dentro de la raíz del proyecto (Ver **Figura 31**).

Figura 31

Código de la página Inicio del administrador web



Nota. Autoría propia.

Una vez con todo listo para poder empezar a crear las funciones de nuestro proyecto, como la lógica del procesamiento y el Frontend, se prosiguió con el desarrollo.

Primero, fue necesario comprender la lógica utilizada en el API que se está implementando, ya que este procesa la imagen (ver **Figura 32**), captura los landmarks faciales y los almacena. Posteriormente, estos datos se compararán con la función que se desarrollará a continuación.

Figura 32

Procesamiento de las imágenes posterior a su subida

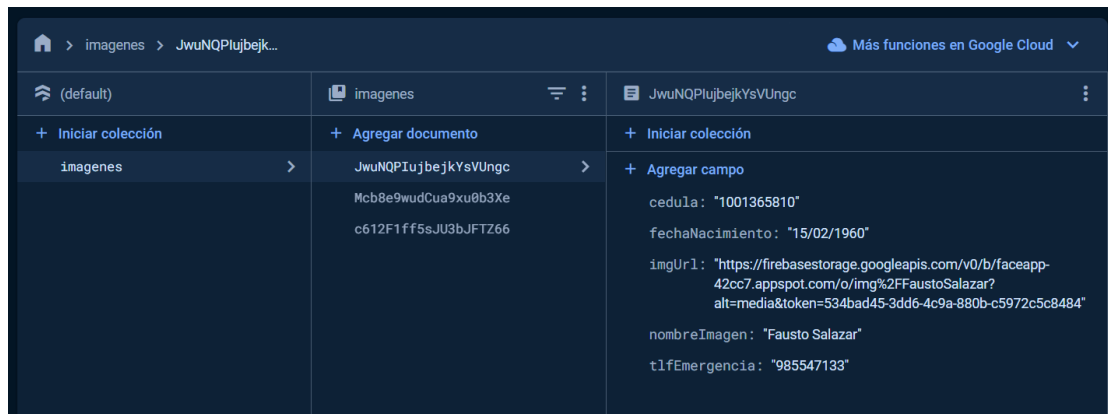
```
TS process-face.service.ts X
src > app > services > TS process-face.service.ts > ProcessFaceService > matchDescriptor
1 import { Injectable } from '@angular/core';
2 import * as faceapi from 'face-api.js';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class ProcessFaceService {
8   idImagen: any;
9   imageDescriptors: any = [];
10  faceMatcher: any;
11
12  constructor() { }
13
14  async processFace(image: any, nombreIm (method) NeuralNetwork<TinyYolov2NetParams>.loadFromUri(uri: string | undefined): Promise<void>
15    await faceapi.nets.tinyFaceDetector.loadFromUri('/assets/models');
16    await faceapi.nets.faceLandmark68Net.loadFromUri('/assets/models');
17    await faceapi.nets.faceRecognitionNet.loadFromUri('/assets/models');
18
19    const detection = await faceapi.detectSingleFace(image, new faceapi.TinyFaceDetectorOptions()
20      .withFaceLandmarks()
21      .withFaceDescriptor());
22
23    if (typeof detection === 'undefined') return;
24
25    this.imageDescriptors.push({
26      id: nombreImagen, // Usar el nombre de la imagen en lugar del ID
```

Nota. Autoría propia.

Posteriormente se realizó la creación de la base de datos, esto se hizo en Firebase, por su facilidad de integración con proyecto dentro de Angular y Node JS (Ver **Figura 33**).

Figura 33

Base de datos alojada en Firebase (Base de datos no relacional)

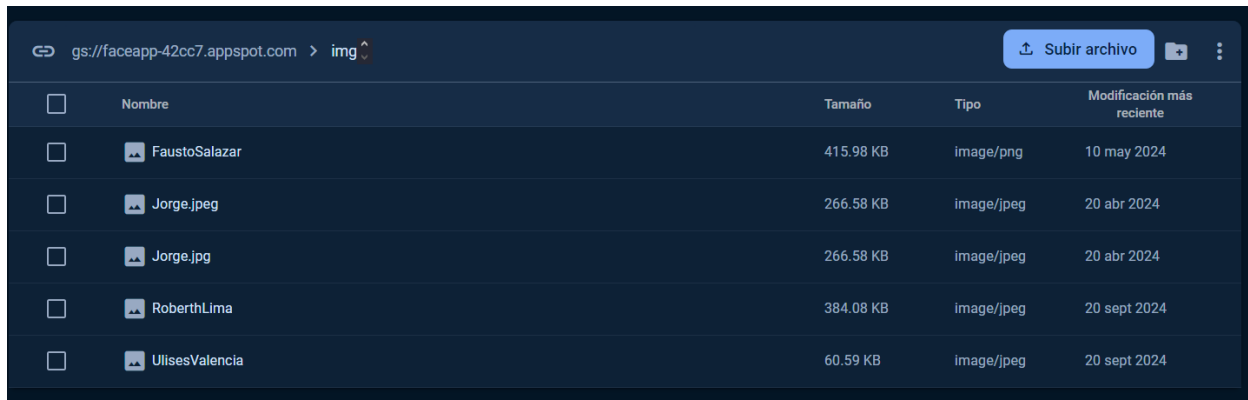


Nota. Autoría propia.

De igual manera se tuvo en consideración esta tecnología debido a que aquí mismo se podía alojar un servidor de contenidos para la subida de imágenes, tal como se muestra en la **Figura 34**, y que esta me devuelva una url de la imagen, y así no saturar dicha base de datos.

Figura 34

Firestore Storage



The screenshot shows the Google Cloud Storage interface for a bucket named 'img'. The URL is 'gs://faceapp-42cc7.appspot.com'. There are six files listed in the table below:

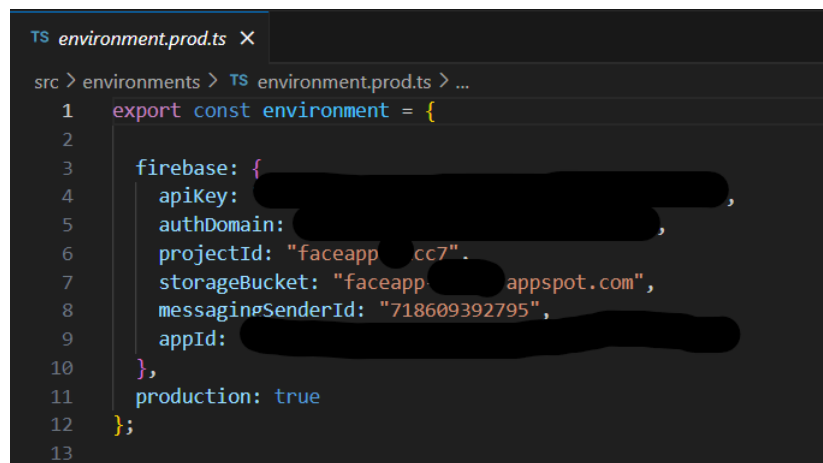
<input type="checkbox"/>	Nombre	Tamaño	Tipo	Modificación más reciente
<input type="checkbox"/>	FaustoSalazar	415.98 KB	image/png	10 may 2024
<input type="checkbox"/>	Jorge.jpeg	266.58 KB	image/jpeg	20 abr 2024
<input type="checkbox"/>	Jorge.jpg	266.58 KB	image/jpeg	20 abr 2024
<input type="checkbox"/>	RoberthLima	384.08 KB	image/jpeg	20 sept 2024
<input type="checkbox"/>	UllsesValencia	60.59 KB	image/jpeg	20 sept 2024

Nota. Autoría propia.

Con todo listo con respecto a la base de datos, se prosiguió con la conexión al proyecto, para que esto fuera de manera directa se utilizaron las credenciales que el mismo Firebase proporciona a sus usuarios (Ver **Figura 35**).

Figura 35

Conexión a la base de datos mediante parámetros y archivo.



```
TS environment.prod.ts X
src > environments > TS environment.prod.ts > ...
1  export const environment = {
2
3    firebase: {
4      apiKey: [REDACTED],
5      authDomain: [REDACTED],
6      projectId: "faceapp-42cc7",
7      storageBucket: "faceapp-42cc7.appspot.com",
8      messagingSenderId: "718609392795",
9      appId: [REDACTED]
10   },
11   production: true
12 };
13
```

Nota. Autoría propia.

Entonces se continuó con el desarrollo de esta, creando 3 páginas de Frontend: Inicio, Subir e Identificación.

Dentro de inicio solo se encuentra un mensaje de bienvenida que explica de que va dicho administrador web, todo codificado en Angular.

Dentro de Subir, se tuvo que crear una interfaz para que permita subir archivos directamente de la computadora, y una tabla que muestre si se ha subido o no la imagen y sus respectivos datos, donde dentro de la tabla se puede eliminar la imagen junto con la información correspondiente, tal como se muestra en la **Figura 36**. También cuenta con un botón que permite recargar los modelos, esto luego de realizar la carga de algunos registros al sistema.

Figura 36

Interfaz gráfica del manejo de los datos por el lado del cliente.



Nota. Autoría propia.

La lógica que se utiliza a la hora de subir una imagen y cargarla con información de esta es como se mencionó antes y como se presenta en la **Figura 38**, la imagen se sube al servidor de contenidos, se genera una url, y se almacena como string dentro de la base de datos, junto con el resto de información crucial de la persona registrada, en la **Figura 37** se indica el código encargado de eso.

Una vez hecho eso se procesa dicha imagen, utilizando los modelos cargados del face-api.js, se identifican y almacenan los landmarks para usarlos en un futuro cuando se vaya a realizar la comparación dando como resultado la identificación facial.

Esto se tiene que realizar con imágenes claras y nítidas de la persona, que se puedan apreciar de manera perfecta sus descriptores faciales, esto con el objetivo de que a la hora de la identificación facial el porcentaje de éxito sea muy por arriba del promedio.

Figura 37

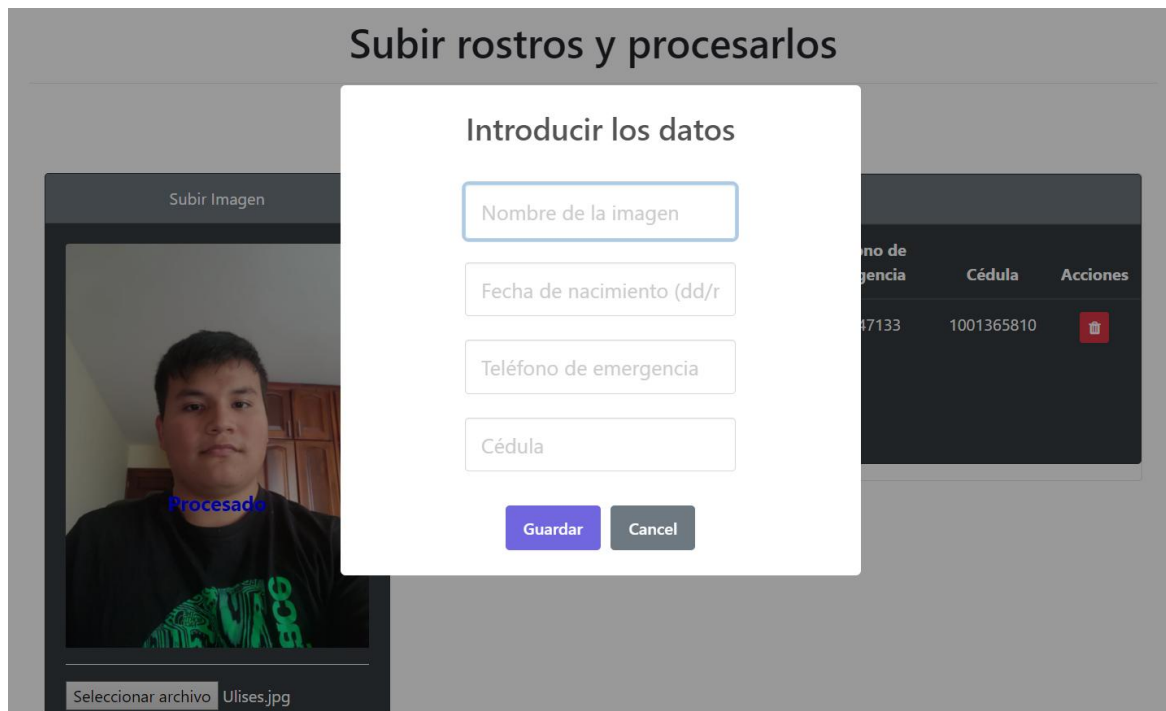
Código encargado de la subida de imágenes e información

```
onSubmit() {  
  
  // Pedir múltiples entradas  
  Swal.fire({  
    title: 'Introducir los datos',  
    html:  
      `      `      `      `    focusConfirm: false,  
    showCancelButton: true,  
    confirmButtonText: 'Guardar',  
    preConfirm: () => {  
      const nombreImagen = (document.getElementById('nombreImagen') as HTMLInputElement).value;  
      const fechaNacimiento = (document.getElementById('fechaNacimiento') as HTMLInputElement).value;  
      const tlfEmergencia = (document.getElementById('tlfEmergencia') as HTMLInputElement).value;  
      const cedula = (document.getElementById('cedula') as HTMLInputElement).value;  
  
      if (!nombreImagen || !fechaNacimiento || !tlfEmergencia || !cedula) {  
        Swal.showValidationMessage('Todos los campos son obligatorios');  
        return false;  
      }  
  
      return {  
        nombreImagen,  
        fechaNacimiento,  
        tlfEmergencia,  
        cedula  
      };  
    }  
  }).then((result) => {
```

Nota. Autoría propia.

Figura 38

Pantalla donde se deben de ingresar los datos



Nota. Autoría propia.

Ahora con respecto a la recarga de los modelos, al iniciar la aplicación, los modelos de FR se cargan automáticamente desde el servidor utilizando face-api.js. Estos modelos, que incluyen redes neuronales para detección de rostros y landmarks (puntos de referencia del rostro), son fundamentales para el correcto funcionamiento del sistema.

El endpoint /reload-models (Ver **Figura 39**) permite recargar estos modelos manualmente si se han añadido nuevas imágenes o si se actualizan las versiones de los modelos, dando como respuesta si se ingresa en el navegador lo que se muestra en la **Figura 40**, y por consola se puede observar la respuesta del servidor (Ver **Figura 41**).

Mientras que por consola en el lado del administrador web muestra un cuadro de diálogo que indica el éxito o fracaso de la operación (Ver **Figura 42**).

Figura 39

Método GET del API para recargar modelos

```

app.get('/reload-models', async (req, res) => {
  res.json({ message: 'Recarga de modelos iniciada' });
  console.log('Recarga de modelos iniciada');

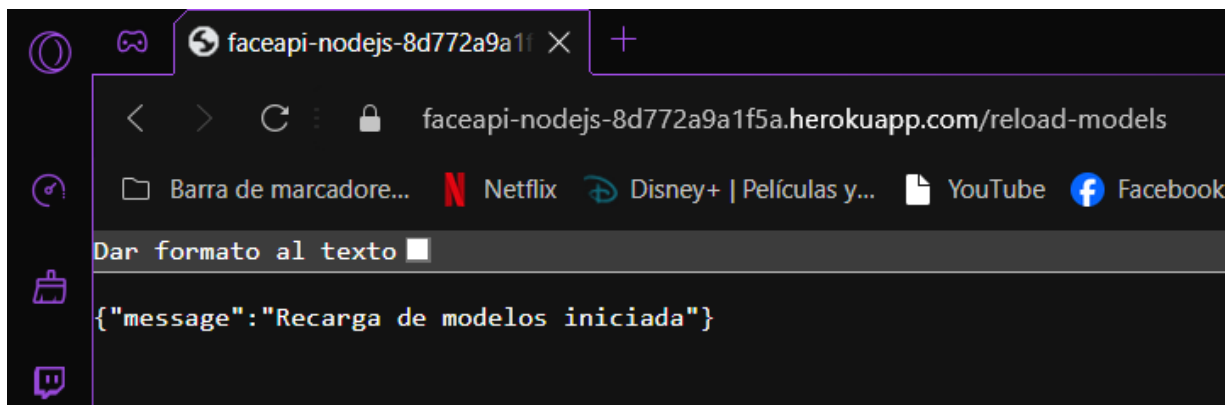
  (async () => {
    try {
      await loadModels();
      await loadLabeledImages();
      console.log('Modelos e imágenes recargados correctamente');
    } catch (err) {
      console.error('Error al recargar los modelos e imágenes:', err);
    }
  })();
});

```

Nota. Autoría propia.

Figura 40

Respuesta del API al recargar modelos (Navegador)



Nota. Autoría propia.

Figura 41

Respuesta del API al recargar modelos (Consola)

```

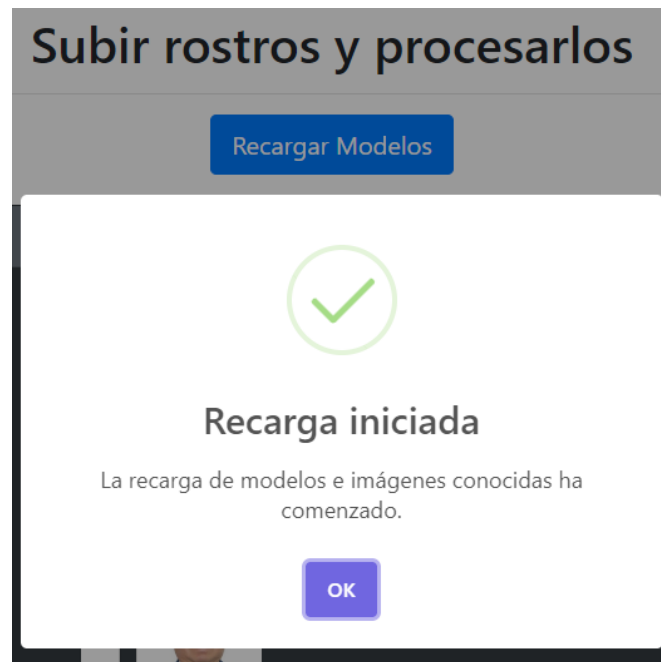
2024-09-22T16:02:15.134079+00:00 app[web.1]: Recarga de modelos iniciada
2024-09-22T16:02:15.719373+00:00 heroku[router]: at=info method=GET path="/favicon.ico" host=faceapi-nodejs-01182a86-6276-4bdd-a455-a6a5dafa798c fwd="157.100.137.219" dyno=web.1 connect=0ms service=310ms status=404
2024-09-22T16:03:21.582205+00:00 app[web.1]: Imágenes y descriptores cargados correctamente desde Firebase
2024-09-22T16:03:21.582288+00:00 app[web.1]: Modelos e imágenes recargados correctamente

```

Nota. Autoría propia.

Figura 42

Prueba del GET reload-models en el cliente



Nota. Autoría propia.

Para eliminar un registro se realizó un método como se muestra en la **Figura 43** que use una función dentro de Firebase, que elimine tanto la imagen, como la información de dicha imagen y que, al realizar la acción dentro del cliente, devuelve un mensaje de éxito, o de error, según la respuesta obtenida (Ver **Figura 44**).

Figura 43

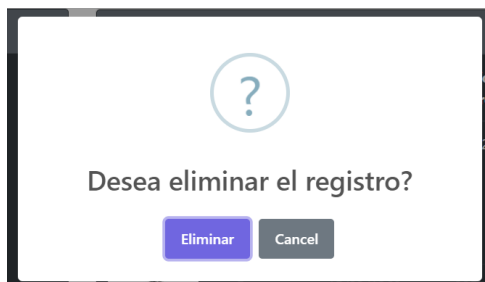
Código responsable de la eliminación de un registro

```
public eliminarImagen(id:string, imagenNombre:string){  
  
    const storage = getStorage();  
  
    const deleteImg = ref(storage, `${this.CARPETA_IMAGENES}/${imagenNombre.replace(/ /g, '')}`);  
  
    deleteObject(deleteImg).then(()=>{  
  
        Swal.fire('EXITO', 'El registro se elimino correctamente', 'success');  
    }).catch((err)=>{  
  
        console.error(err);  
    });  
  
    return this.imagenesCollection.doc(id).delete();  
}
```

Nota. Autoría propia.

Figura 44

Pantalla de confirmación al eliminar un registro



Nota. Autoría propia.

Se presenta en la **Tabla 20**, la reunión de retrospectiva del Sprint, que recopila los aspectos positivos, aspectos negativos, lecciones aprendidas y las acciones propuestas. Este documento es fundamental para la mejora continua del equipo y el proceso de desarrollo.

Tabla 20

Reunión de retrospectiva Sprint 0

Reunión de retrospectiva				
Sprint: 0				
Fecha: 31/07/2024				
Actividad / Historia de usuario	Aspectos positivos	Aspectos negativos	Lecciones aprendidas	Acciones propuestas
HU-01 Subir imágenes de estudiantes al administrador web	Interfaz sencilla e intuitiva, fácil de manejar por parte del administrador	Dependencia del modelo cargado local	Una interfaz siempre debe ser lo más simple y funcional posible	Mejorar el aspecto visual y verificar una forma de optimización
HU-02 Recargar los modelos de reconocimiento facial	Posibilidad de subir varias imágenes y luego tener la opción de recarga	Demora a la hora de realizar la acción	En aplicación que consumen API es importante usar memoria cache y procesos en segundo plano para no saturar al servidor	Indagar la demora del proceso de recarga
HU-03 Visualizar y administrar imágenes junto sus datos	Mensaje correcto a la hora de querer eliminar un registro	Dependencia de la rapidez de Firebase	El manejo de datos siempre se debe controlar con un mensaje de confirmación	Devolver un mensaje de que el registro ha sido eliminado exitosamente

Nota. Autoría propia.

2.3.3. Sprint 1

A continuación, en la **Tabla 21**, se presenta la tabla del Sprint Backlog del Sprint 1, que detalla las historias de usuario y las tareas asignadas para el presente sprint, así como las estimaciones de tiempo correspondientes. Este documento es esencial para la gestión y seguimiento del avance del proyecto.

Tabla 21

Sprint 1

SPRINT BACKLOG				
Sprint: 1				
Total horas: 66				
Fecha Inicio SP1: 01/08/2024				
Fecha Final SP1: 31/08/2024				
Actividad / Historia de usuario	Fase Desarrollo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO REAL (Horas)
HU-04 Identificar una persona en tiempo real	Desarrollo	Diseño de la interfaz para el video el tiempo real	3	4
	Desarrollo	Implementación de la lógica para activar la cámara	5	6
	Desarrollo	Desarrollo de la funcionalidad para capturar imágenes	15	20
	Desarrollo	Integración de FaceAPI.js para la identificación en tiempo real	25	30
	Desarrollo	Implementación de la lógica para mostrar resultados de identificación	10	3
	Desarrollo	Pruebas y ajustes	8	5
Reuniones	Gestión	Planificación	0.2	0.2
	Gestión	Diarias	0.1	0.1
	Gestión	Revisión	0.1	0.1
	Gestión	Retrospectiva	0.1	0.1
TOTAL			66	68

Nota. Autoría propia.

En el Sprint 1, se desarrolla y se revisa el cumplimiento de toda la funcionalidad que conlleva la identificación facial, ya que, al probar esta funcionalidad, de manera indirecta también se verifican las funcionalidades anteriores: Subida de imágenes e información, la

eliminación de registros y la recarga de los nuevos modelos subidos en la página de Subir. Todo eso previamente establecido en **Tabla 2** hasta la **Tabla 4**, donde se encuentran las historias de usuario correspondientes.

Se comprobó el completo flujo del sistema, para asegurar que los módulos del administrador web estuvieran alineados con la funcionalidad de reconocimiento e identificación facial. Esto permitió garantizar que el sistema pudiera interactuar adecuadamente con la cámara y realizar las comparaciones con la base de datos y los landmarks almacenados al momento de subir un nuevo registro, y de recargar los modelos.

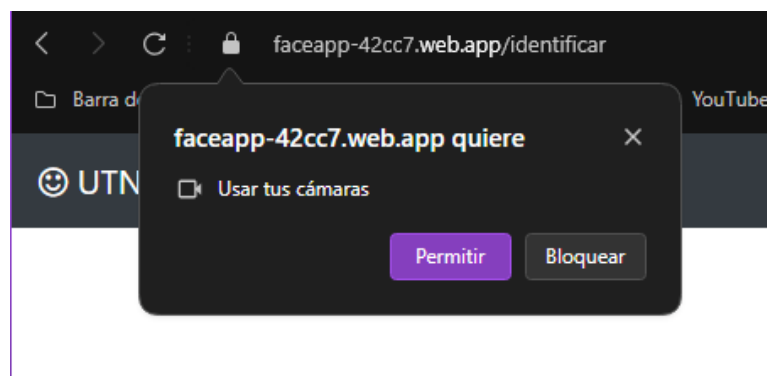
Se probó el acceso a la cámara del navegador en diferentes dispositivos y configuraciones de seguridad para asegurar que los usuarios puedan otorgar permisos correctamente y activar la cámara sin inconvenientes (Ver **Figura 45**).

De igual forma se integró la funcionalidad que permite al administrador web activar la cámara del dispositivo, confirmando que el administrador podía acceder a la imagen en tiempo real (Ver **Figura 47**).

La **Figura 48** muestra el resultado, que es un cuadro en donde se presenta el video capturado por la cámara de la laptop en tiempo real.

Figura 45

Solicitud de aprobación de acceso a la cámara



Nota. Autoría propia.

Para la captura de la imagen en tiempo real se codifico un método que permita crear un canvas, dentro del cual se pueda “dibujar” un recuadro, y que dentro de este se presente el video en tiempo real capturado con la cámara del dispositivo (Laptop) dentro del navegador, se muestra el código en la **Figura 46**.

Figura 46

Creación del canvas Activación de la cámara web

```
})
export class IdentificarComponent implements OnInit, OnDestroy {
  @ViewChild('videoContainer', { static: true }) videoContainer!: ElementRef;
  @ViewChild('myCanvas', { static: true }) myCanvas!: ElementRef;

  imagenes: any[] = [];
  public context!: CanvasRenderingContext2D;
  labeledDescriptors: any[] = [];
  detectedName: string = '';
  detectedBirthdate: string = ''; // Nueva propiedad
  detectedEmergencyContact: string = ''; // Nueva propiedad
  detectedId: string = ''; // Nueva propiedad
  private videoStream: MediaStream | null = null;
}
```

Nota. Autoría propia.

Figura 47

Activación de la cámara web

```
ngOnInit(): void {
  this.startCamera();
}

ngOnDestroy(): void {
  this.stopCamera();
}

startCamera = async () => {
  this.context = this.myCanvas.nativeElement.getContext('2d');
  this.videoStream = await navigator.mediaDevices.getUserMedia({ video: true });
  const stream = this.videoContainer.nativeElement;
  stream.srcObject = this.videoStream;

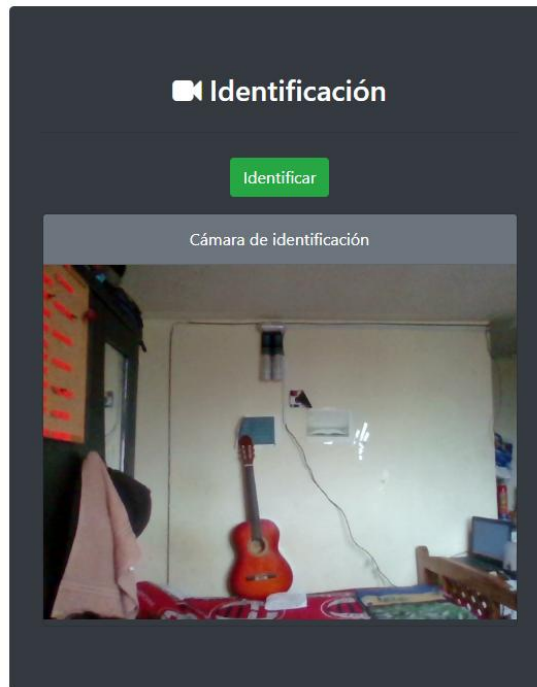
  const reDraw = async () => {
    this.context.drawImage(stream, 0, 0, 640, 480);
    requestAnimationFrame(reDraw);
  }

  requestAnimationFrame(reDraw);
}
```

Nota. Autoría propia.

Figura 48

Presentación en el Administrador web del contenedor de video



Nota. Autoría propia.

Una vez aprobado todos los requisitos anteriores se pudo comenzar con la lógica que va a haber detrás de la identificación facial, y la muestra de los datos. El proceso de identificación se inicia llamando al método `deteccion()`, que carga los modelos pre-entrenados y empieza el FR mediante la librería `face-api.js`. Primero, se cargan los modelos necesarios para la detección, se muestra como en la **Figura 49**, desde la carpeta de modelos.

Figura 49

Carga de los modelos de `face-api.js`

```
loadModelsAndDetect = async () => {  
  await faceapi.nets.tinyFaceDetector.loadFromUri('/assets/models');  
  await faceapi.nets.faceLandmark68Net.loadFromUri('/assets/models');  
  await faceapi.nets.faceRecognitionNet.loadFromUri('/assets/models');
```

Nota. Autoría propia.

Una vez que los modelos están cargados, la aplicación comienza a detectar rostros dentro del video en tiempo real. El método processFace utiliza face-api.js para detectar un solo rostro en cada frame del video.

Si no se detecta ningún rostro, el método simplemente retorna. Si se detecta un rostro, se obtiene el descriptor facial que es una representación matemática única del rostro detectado.

El descriptor facial obtenido se compara con los descriptores previamente almacenados, utilizando el servicio ProcessFaceService, si se encuentra una coincidencia, se obtiene el nombre de la persona junto con otros datos relevantes, como la fecha de nacimiento, el contacto de emergencia y la cédula de identidad. Estos datos son mostrados en pantalla y dibujados en el canvas justo encima del cuadro de detección del rostro (Ver **Figura 50**).

Figura 50

Verificación de identificación facial



Nota. Autoría propia.

El proceso de reconocimiento se ejecuta de manera periódica cada dos segundos, utilizando setInterval.

Para realizar la identificación, los descriptores de las imágenes almacenadas previamente son cargados a través del servicio `ImagenesService`. Este servicio obtiene una lista de imágenes almacenadas y procesa cada una, asociándolas con los descriptores faciales (Ver **Figura 51**).

Figura 51

Código final encargado de la detección y muestra de información

```

imagesLista() {
  this.imagenesSvc.getImagenes().subscribe((res: any) => {
    this.imagenes = res;
    this.labeledDescriptors = this.imagenes.map((imagen: any) => {
      const imageElement = document.createElement('img');
      imageElement.src = imagen.imgUrl;
      imageElement.crossOrigin = 'anonymous';
      return this.processSvc.processFace(imageElement, imagen.nombreImagen, imagen.fechaNacimiento, imagen.tlfEmergencia, imagen.cedula);
    });
  });
}

```

Nota. Autoría propia.

De este modo, se establece una correspondencia entre los rostros detectados en tiempo real y los datos almacenados en la base de datos.

Se presenta en la **Tabla 22**, la reunión de retrospectiva del Sprint, que recopila los aspectos positivos, aspectos negativos, lecciones aprendidas y las acciones propuestas. Este documento es fundamental para la mejora continua del equipo y el proceso de desarrollo.

Tabla 22

Reunión de retrospectiva Sprint 1

Reunión de retrospectiva				
Sprint: 1				
Fecha: 31/08/2024				
Actividad / Historia de usuario	Aspectos positivos	Aspectos negativos	Lecciones aprendidas	Acciones propuestas
HU-03 Identificar una persona en tiempo real	Rápido manejo con relación a las comparaciones de imágenes en la base de datos con la web	Recarga cada 2 segundos, llamado excesivo del face-api.js	Los permisos de la cámara son un poco sensibles por la privacidad	Revisar si es conveniente un intervalo tan corto a la hora de identificar

Nota. Autoría propia.

2.3.4. Sprint 2

A continuación, en la **Tabla 23**, se presenta la tabla del Sprint Backlog del Sprint 2, que detalla las historias de usuario y las tareas asignadas para el presente sprint, así como las estimaciones de tiempo correspondientes. Este documento es esencial para la gestión y seguimiento del avance del proyecto.

Tabla 23

Sprint 2

SPRINT BACKLOG				
Sprint: 2				
Total horas: 96				
Fecha Inicio SP1: 01/09/2024				
Fecha Final SP1: 11/11/2024				
Actividad / Historia de usuario	Fase Desarrollo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO REAL (Horas)
Creación del proyecto	Desarrollo	Creación y configuración del proyecto	6	5
HU-06 Identificar una persona y mostrar su información en realidad aumentada	Desarrollo	Desarrollo de la API en Node.js para la identificación	30	20
	Desarrollo	Implementación de la lógica para consumir la API en Unity	15	10
	Desarrollo	Configuración de AR Foundation en el proyecto de Unity	20	30
	Desarrollo	Implementación de la visualización de información en AR	20	25
	Desarrollo	Pruebas y ajustes	5	4
Reuniones	Gestión	Planificación	0.5	0.5
	Gestión	Diarias	0.1	0.1
	Gestión	Revisión	0.1	0.1
	Gestión	Retrospectiva	0.1	0.1
TOTAL			96	94

Nota. Autoría propia.

Continuando ahora con el Sprint 2, y antes que todo, se empieza con la creación la API, la cual será necesario para luego consumirse dentro del aplicativo móvil y lograr la detección, reconocimiento e identificación facial. Que permita la identificación de personas.

Esto se realizó a través de la API construida en Node.js, que además de manejar la carga de imágenes y datos, controla la lógica del reconocimiento y la comparación en tiempo real. Dentro de este código se descargó de manera local los modelos que ocupa face-api.js para el reconocimiento de rostros, y no depender al 100% de la capacidad de respuesta que el API en línea pueda brindar.

Ya con los requisitos terminados con respecto al FR se va a proseguir con la conexión directa a Firebase (Ver **Figura 52**), la cual contiene tanto la base de datos (Firestore Database) y el servidor de contenidos (Storage), esto para manejar el almacenamiento de los datos personales y las imágenes que se utilizan en el proceso de identificación facial.

Firebase fue elegido base de datos en este proyecto debido a su facilidad de uso, escalabilidad y la integración nativa con tecnologías basadas en la nube, lo que es ideal para aplicaciones móviles y web. Para ello se configuró el SDK de Firebase en el proyecto de Node.js. Se generaron las credenciales necesarias en la consola de Firebase, específicamente el archivo serviceAccountKey.json, que contiene las credenciales para la autenticación segura y acceso a los servicios de Firebase.

Figura 52

Conexión a la base de datos Firebase

```
const admin = require('firebase-admin');
const serviceAccount = require('./serviceAccountKey.json');

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
  databaseURL: 'https://<your-project-id>.firebaseio.com'
});

const db = admin.firestore();
```

Nota. Autoría propia.

En este paso, se inicializó Firebase Admin SDK, que permitió interactuar con Firebase Firestore (base de datos NoSQL) y Firebase Storage (para almacenar las imágenes de los usuarios).

Cuando un usuario registra una nueva imagen, se sube a Firebase Storage. Esto permite un almacenamiento centralizado y eficiente de las imágenes, las cuales son luego accesibles desde cualquier parte de la aplicación. Para la subida de imágenes, se utilizó el bucket de Firebase Storage, la cual sube las imágenes al bucket de Firebase y genera una URL de acceso que se asocia con los datos del usuario en Firestore.

Los datos personales de los usuarios identificados, como el nombre, fecha de nacimiento, teléfono de emergencia y cédula, se almacenan en Firebase Firestore. Esta base de datos NoSQL permite almacenar documentos que contienen campos clave-valor, lo que facilita la asociación de datos adicionales a cada imagen registrada.

Cuando se registra un nuevo usuario, se crea un documento en la colección users de Firestore. Cada documento está identificado por un ID único (generalmente el nombre o número de cédula de la persona) y contiene campos como el URL de la imagen subida, la fecha de nacimiento, el teléfono de emergencia, entre otros.

Durante el proceso de identificación facial, una vez que se detecta una coincidencia con un rostro almacenado en el sistema, los datos correspondientes del usuario se recuperan desde Firestore (Ver **Figura 53**). Esto permite mostrar información adicional relevante a la persona identificada.

Figura 53

Respuesta de Firebase con los datos

```
const getUserData = async (userId) => {
  const userRef = db.collection('users').doc(userId);
  const doc = await userRef.get();
  if (!doc.exists) {
    throw new Error('No se encontró el usuario en la base de datos');
  }
  return doc.data();
};
```

Nota. Autoría propia.

A través de la ruta /upload (Ver **Figura 54**), el sistema permite que los usuarios suban imágenes para realizar el proceso de identificación facial. Utilizando multer, la imagen se carga en memoria, asegurando que no se exceda el tamaño máximo permitido de 5 MB. Una vez cargada, la imagen es procesada para detectar rostros y compararlos con los modelos cargados.

Figura 54

Método POST del API para subir una imagen

```
app.post('/upload', upload.single('image'), async (req, res) => {
  if (!modelsLoaded || !labeledImagesLoaded) {
    return res.status(503).json({ error: 'Modelos e imágenes aún no están cargados' });
  }

  if (!req.file) {
    return res.status(400).json({ error: 'No file uploaded' });
  }

  try {
    const result = await processImage(req.file.buffer);
    res.json(result);
  } catch (err) {
    res.status(500).json({ error: 'Error procesando la imagen' });
  }
});
```

Nota. Autoría propia.

Esta funcionalidad se complementa con el uso de una cache en memoria que almacena los resultados del procesamiento de imágenes. Esto permite mejorar el rendimiento cuando se intenta identificar una imagen que ya ha sido procesada anteriormente.

Continuando con la funcionalidad principal, para detectar todos los rostros presentes en la imagen subida, la función processImage (Ver **Figura 55**) utiliza face-api.js. Luego, los descriptores de los rostros son comparados con los descriptores almacenados de personas previamente registradas. Si hay coincidencia, se retorna la información adicional de la persona (como fecha de nacimiento, teléfono de emergencia, cédula, entre otros).

Figura 55

Código encargado de procesar las imágenes

```

const processImage = async (imageBuffer) => {
  const img = await canvas.loadImage(imageBuffer);
  const detections = await faceapi.detectAllFaces(img).withFaceLandmarks().withFaceD

  if (detections.length === 0) {
    return { error: 'No se detectaron rostros en la imagen' };
  }

  const results = detections.map(d => {
    const bestMatch = faceMatcher.findBestMatch(d.descriptor);
    const match = labeledDescriptors.find(ld => ld.label === bestMatch.label);
    return {
      label: match.label,
      fechaNacimiento: match.fechaNacimiento,
      tlfEmergencia: match.tlfEmergencia,
      cedula: match.cedula,
      distance: bestMatch.distance
    };
  });

  return { facesDetected: detections.length, matches: results };
};

```

Nota. Autoría propia.

Para un mejor entendimiento se describen los servicios disponibles codificados para el API, los cuales sirven para la integración con Unity y AR Foundation con la detección facial/FR. La ruta base para consumir los servicios es: <https://faceapi-nodejs-8d772a9a1f5a.herokuapp.com>, a esta se le tiene que agregar diferencias rutas finales o sufijos de la URL dependiendo de la función que se quiera consumir.

A continuación, en la **Tabla 24** se detallan cada uno de los servicios creados y disponibles del API.

Tabla 24*Catálogo de servicios REST*

Catálogo de servicios REST		
Ruta base	https://faceapi-nodejs-8d772a9a1f5a.herokuapp.com	
Ruta del servicio	Tipo de petición	Descripción
/upload	POST	Sube una imagen para su procesamiento y FR. Devuelve resultados procesados.
/users	GET	Obtiene la lista de usuarios (rostros) almacenados en el sistema.
/reload-models	GET	Inicia la recarga de modelos e imágenes conocidas. Responde con un mensaje de inicio de recarga.

Nota. Autoría propia.

Este desarrollo permitió que el sistema pudiera identificar y comparar rostros de manera efectiva, cargando y actualizando los modelos cuando fuera necesario, garantizando un flujo continuo desde la subida de una nueva imagen hasta su procesamiento y verificación.

Continuando con el desarrollo, para identificar a una persona y mostrar su información utilizando AR, se utilizó AR Foundation en Unity, junto con una API personalizada desarrollada en Node.js, la cual consume face-api.js para realizar el FR.

Este sprint fue fundamental, ya que permitió integrar la tecnología principal en la aplicación móvil, mostrando los datos personales de la persona identificada sobre un objeto tridimensional mediante AR.

El primer paso para la identificación facial fue implementar la captura de una imagen en tiempo real desde la cámara AR del dispositivo. El componente ARCameraManager fue configurado para tomar la imagen cuando el usuario lo solicite, utilizando una función llamada TakePhoto. Aquí se inicia una captura de pantalla de lo que la cámara AR está viendo,

transformando la imagen en una textura que posteriormente será enviada a la API para su análisis.

Una vez capturada la imagen, esta se convierte en un array de bytes utilizando el método `EncodeToPNG()` y se envía a la API personalizada que consume *face-api.js* (Ver **Figura 56**). Esta API, desarrollada en Node.js, procesa la imagen y compara el rostro capturado con los modelos faciales almacenados. En caso de encontrar una coincidencia, la API devuelve los datos personales de la persona identificada.

Figura 56

Código encargado de consumir el POST del API (Procesar y comparar)

```
private IEnumerator SendPhotoToAPI(Texture2D texture) {
    byte[] imageBytes = texture.EncodeToPNG();
    WWWForm form = new WWWForm();
    form.AddBinaryData("image", imageBytes, "photo.png", "image/png");
    UnityWebRequest request = new UnityWebRequest(apiUrl, UnityWebRequest.kHttpVerbPOST);
    request.uploadHandler = new UploadHandlerRaw(form.data);
    request.downloadHandler = new DownloadHandlerBuffer();
    yield return request.SendWebRequest();
}
```

Nota. Autoría propia.

Una vez la API responde con los datos del rostro reconocido, la aplicación extrae la información clave de la persona, como el nombre, fecha de nacimiento, teléfono de emergencia y cédula, mostrándola en tiempo real mediante AR. Para esto, se utilizó una interfaz en Unity compuesta por un objeto 3D al que se le agregó un Canvas con un `TextMeshProUGUI`, que es donde se despliega la información (Ver **Figura 57**).

Figura 57

Código encargado de consumir el POST del API (Respuesta del procesado y comparativa)

```
try {
    if (apiResponse.matches.Count > 0) {
        Match firstMatch = apiResponse.matches[0];
        arFaceMan.facePrefab.transform.Find("Cube").transform.Find("Canvas").transform
            "Nombre: " + firstMatch.label + "\n" +
            "Fecha de Nacimiento: " + firstMatch.fechaNacimiento + "\n" +
            "Teléfono de Emergencia: " + firstMatch.tlfEmergencia + "\n" +
            "Cédula: " + firstMatch.cedula + "\n" +
            "Distancia: " + firstMatch.distance.ToString();
    }
}
```

Nota. Autoría propia.

La integración con AR Foundation permitió que el sistema de FR no solo identificara a las personas, sino que también presentara la información de una manera intuitiva mediante la superposición de textos sobre el rostro detectado en un entorno AR.

Este enfoque brinda una experiencia interactiva, donde los usuarios pueden ver en tiempo real los datos correspondientes a la persona reconocida mientras interactúan con su entorno a través de la cámara.

Se implementaron mecanismos para manejar errores, tal como se indica en la **Figura 58**), como cuando no se detecta un rostro válido o no se encuentra una coincidencia en la base de datos. En estos casos, se muestra un mensaje adecuado en el objeto AR. Además, el sistema se asegura de reactivar el rastreo facial luego de cada identificación para mantener el ciclo de detección continuo.

Figura 58

Manejo de errores al consumir el POST del API

```
// Verifica si hubo errores durante el envío de la petición
if (request.result == UnityWebRequest.Result.ConnectionError || request.result == UnityWebRequest.Result.ProtocolError)
{
    arFaceMan.facePrefab.transform.Find("Cube").transform.Find("Canvas").transform.Find("Text (TMP)").GetComponent<TextMeshPro>().text = "No se encontró un rostro válido";
    Debug.LogError("Error en la petición POST: " + request.error);
}
```

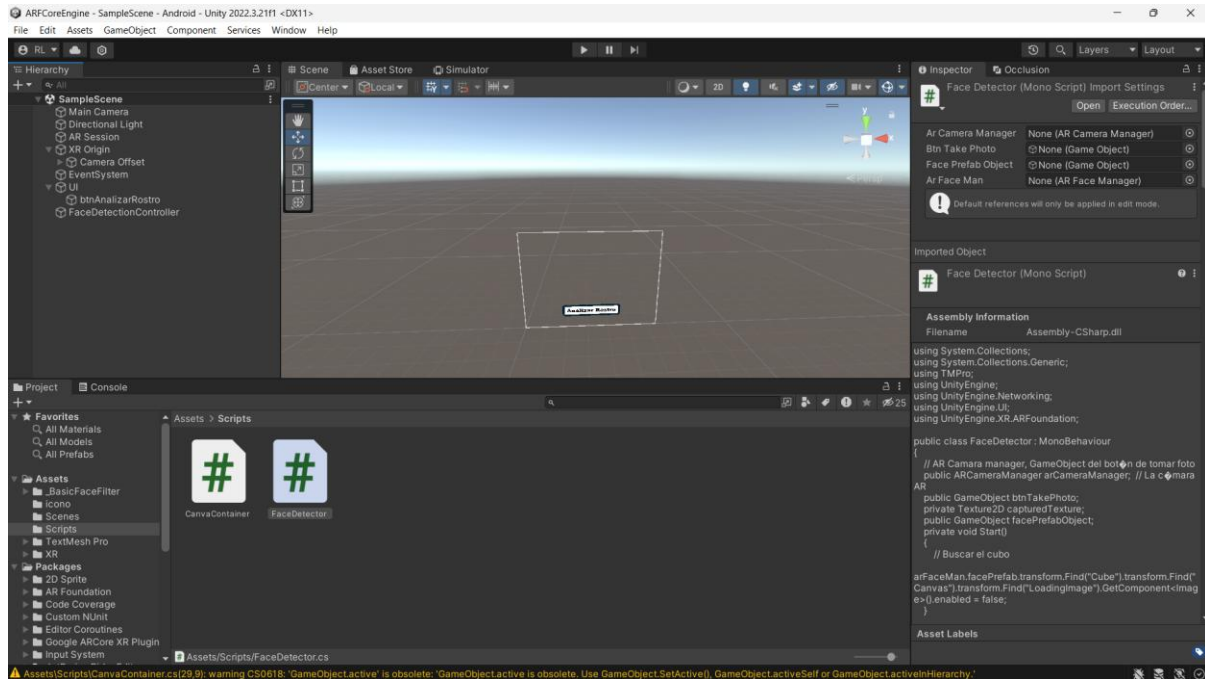
Nota. Autoría propia.

Finalmente, se compiló la aplicación móvil utilizando Unity (Ver **Figura 59**), integrando todas las funcionalidades descritas. Esto permitió empaquetar el proyecto

para dispositivos móviles, asegurando que la aplicación fuera completamente operativa en cuanto a la identificación facial y la visualización de datos mediante AR.

Figura 59

Proyecto finalizado en Unity



Nota. Autoría propia.

La interfaz dentro del aplicativo móvil es lo más sencillo e intuitivo posible, contando únicamente con un botón, el cual realizar el análisis facial, y muestra la información en caso de que la comparación sea exitosa y la persona en cuestión se encuentre cargada en la base de datos (Ver **Figura 60**).

Una vez la aplicación haya reconocido el rostro y si este se encuentra registrado en la base de datos, la información será presentada con AR, como se observa en la **Figura 61**.

Figura 60

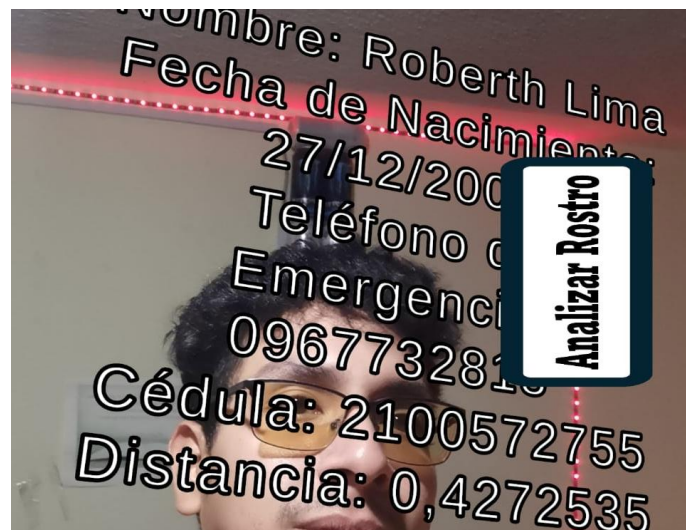
Aplicación móvil



Nota. Autoría propia.

Figura 61

Presentación de los datos con realidad aumentada



Nota. Autoría propia.

En conclusión, este sprint consolidó el uso de la AR y el FR en la aplicación móvil, permitiendo la identificación en tiempo real de personas y la visualización de sus datos personales de manera intuitiva y efectiva.

Se presenta en la **Figura 25**, la reunión de retrospectiva del Sprint, que recopila los aspectos positivos, aspectos negativos, lecciones aprendidas y las acciones propuestas. Este documento es fundamental para la mejora continua del equipo y el proceso de desarrollo.

Tabla 25

Reunión de retrospectiva Sprint 3

Reunión de retrospectiva				
Sprint: 3				
Fecha: 11/11/2024				
Actividad / Historia de usuario	Aspectos positivos	Aspectos negativos	Lecciones aprendidas	Acciones propuestas
HU-05 Identificar una persona y mostrar su información en realidad aumentada	Interfaz sencilla e intuitiva, fácil de manejar por parte del administrador	Dependencia del modelo cargado local	Una interfaz siempre debe ser lo más simple y funcional posible	Mejorar el aspecto visual y verificar una forma de optimización

Nota. Autoría propia.

CAPÍTULO 3

3.1. Verificación de resultados

La evaluación de cumplimiento se utilizó como método para validar los resultados, basándose en la característica de satisfacción, respecto a la Calidad en Uso de la ISO/IEC 25010, específicamente acorde con la subcaracterística de utilidad (ISO 25010, 2022).

3.1.1. Medición del modelo de calidad en uso

- **Información del sistema**

En la Figura 62, se especifican los datos necesarios para detallar el producto de software y realizar la evaluación.

Figura 62

Matriz de calidad de Software

MATRIZ DE CALIDAD DE SOFTWARE		
1. DATOS INFORMATIVOS:		
Fecha:	11/12/2024	
Institución:	Universidad Técnica del Norte	
Nombre del Software:		
OBJETIVOS GENERALES DEL SOFTWARE		
Desarrollar una aplicación móvil para reconocimiento facial con realidad aumentada utilizando Unity y AR Foundation que permita la identificación de personas dentro del campus de la Universidad Técnica del Norte.		
OBJETIVOS ESPECÍFICOS DEL SOFTWARE		
<ul style="list-style-type: none"> • Establecer un marco teórico referente al desarrollo de aplicaciones móviles con reconocimiento facial y realidad aumentada. • Construir la aplicación móvil junto a su administrador web utilizando la metodología ágil Scrum • Validar los resultados usando la norma ISO 25010 evaluación de calidad en uso aplicando la característica de satisfacción con la subcaracterística de utilidad. 		
PARTICIPANTES		
Cargo	Nombre	Unidad
Product Owner	Sr. Roberth Gabriel Lima	Responsable del desarrollo del sistema de reconocimiento facial y realidad aumentada.
Scrum Master	MSc. Fausto Salazar	Director del trabajo de titulación
Desarrollador	Sr. Roberth Gabriel Lima	Tesista

Nota. Adaptado de (Vaca, 2017).

- **Elección del tipo del producto de Software**

En la **Figura 63**, se define el tipo de producto de software que se evaluará.

Figura 63

Tipo de Software

2. TIPO DE PRODUCTO SOFTWARE		
Producto	Clasificación de producto	Selección
Página Web (PW)	Estática	
	Animada	
	Dinámica	
	Portal Web	
	Tienda Virtual o Comercio Electrónico	
	Página Web con Gestor de Contenido	
	Página Web 2.0	
Base de Datos (BDD)	BDD jerárquica	
	BDD de red	
	BDD transaccional	
	BDD relacional	
	BDD multidimensional	
	BDD orientado a objetos	
	BDD documental	
Software de Aplicación (SA)	BDD deductiva	
	SA de productividad (editores de texto)	
	SA de entretenimiento (videojuegos)	
	SA de negocios (ERP)	
	SA de educación (programas interactivos de aprendizaje)	
	SA de tecnología (control de sistemas, médicas, etc.)	X

Nota. Adaptado de (Vaca, 2017).

- **Característica y subcaracterística establecida**

En la **Figura 64**, se modificó la matriz de calidad en uso para la sección de la respectiva característica y subcaracterística, la cual es Satisfacción, y utilidad, respectivamente.

Figura 64

Característica y subcaracterística

8. SUBCARACTERÍSTICAS DE CALIDAD EN USO				
Característica	Subcaracterística	Nivel de Importancia	%	Total Característica
C1 - Efectividad	Efectividad	No Aplica		0%
C2 - Eficiencia	Eficiencia	No Aplica		0%
C3 - Satisfacción	Utilidad	Alta	100%	100%
C4 - Libertad de Riesgo	Libertad del riesgo económico	No Aplica		0%
	Libertad del riesgo de salud y seguridad	No Aplica		
	Libertad del riesgo ambiental	No Aplica		
C5 - Cobertura de contexto	Complejidad de Contexto	No Aplica		0%
	Flexibilidad	No Aplica		

Nota. Adaptado de (Vaca, 2017).

3.2. Análisis e interpretación de resultados

3.2.1. Especificación de muestra

Los datos se recolectaron mediante un muestreo no probabilístico por conveniencia, eligiendo a un grupo de estudiantes que cursan la carrera de software que completaron la encuesta luego de probar la aplicación móvil desarrollada.

3.2.2. Desarrollo de encuesta

Para medir la satisfacción del usuario, se optó por la versión número del Computer System Usability Questionnaire (CSUQ), enfocándose en la subcaracterística de utilidad.

CSUQ se destaca como una herramienta para determinar el nivel de satisfacción de los usuarios, la utilidad percibida de los sistemas informáticos. El cuestionario aborda directamente estos aspectos, proporcionando datos valiosos sobre cómo los usuarios perciben la Comodidad, confianza y satisfacción al interactuar con el sistema. Esta capacidad de evaluación integral respalda esta elección para estudios centrados en la satisfacción del usuario y la utilidad del sistema (Lewis, 1995).

CSUQ evalúa 4 factores:

- a) Calidad del sistema: conformado por las preguntas 1 a la 6
- b) Calidad de la información: conformado por las preguntas 7 a la 12
- c) Calidad de la interfaz: conformado por las preguntas 13 a la 15
- d) Satisfacción general: conformado por la pregunta 16

A continuación, se muestran las 16 preguntas de la tercera versión del CSUQ propuestas por (Lewis, 2018):

- a) En general, estoy satisfecho con lo fácil que es usar este sistema.
- b) Fue sencillo usar este sistema.
- c) Pude completar mi trabajo rápidamente usando este sistema.
- d) Me sentí cómodo usando este sistema.

- e) Fue fácil aprender a usar este sistema.
- f) Creo que podría ser productivo rápidamente usando este sistema.
- g) El sistema dio mensajes de error que me indicaron claramente cómo solucionar problemas.
- h) Cada vez que cometía un error al utilizar el sistema, podía recuperarme fácil y rápidamente.
- i) La información (como ayuda en línea, mensajes en pantalla y otra documentación) provista con este sistema era clara.
- j) Fue fácil encontrar la información que necesitaba.
- k) La información provista por el sistema fue efectiva para ayudarme a completar mi trabajo.
- l) La organización de la información en las pantallas del sistema fue clara.
- m) La interfaz de este sistema fue agradable.
- n) Me gustó usar la interfaz de este sistema.
- o) Este sistema tiene todas las funciones y capacidades que espero que tenga.
- p) En general, estoy satisfecho con este sistema.

Posteriormente, se vincularon las preguntas previas con las métricas de la característica de Satisfacción y subcaracterística de Utilidad, de la norma ISO/IEC 25010. (Ver **Tabla 26**).

Tabla 26

Relación de las métricas de Satisfacción-Utilidad de la norma ISO 25010 con las preguntas de CSUQ

Métricas	Preguntas CSUQ
Satisfacción	a, p, c, o, n, m
Confianza	g, h, i, j, l, k
Comodidad	d, e, f, b

Nota. Autoría propia.

Esto emplea una escala Likert de 7 puntos, donde 1 corresponde a totalmente de acuerdo y 7 a totalmente en desacuerdo, como se aprecia en la **Tabla 27**.

Tabla 27

Respuestas escala Likert de 7 puntos

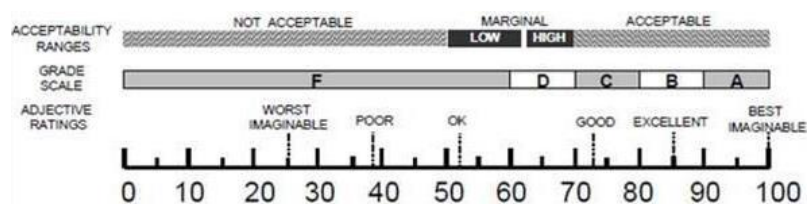
Opción de respuesta	Valor/Escala
Totalmente de acuerdo	1
Bastante de acuerdo	2
De acuerdo	3
Neutral	4
En desacuerdo	5
Bastante en desacuerdo	6
Totalmente en desacuerdo	7

Nota. Adaptado de (Fabila et al., 2014).

Hay un cuestionario aparte, utilizado para calcular la experiencia de los usuarios con un software, System Usability Scale (SUS). Sin embargo, para el análisis de la satisfacción y utilidad del software en esta investigación, se optó por adaptar los principios de SUS. De este cuestionario, se extrajo su escala de valores presentada en la Figura 65, con el fin de evaluar cómo los usuarios perciben tanto la satisfacción general con la aplicación como la utilidad que perciben de las funciones ofrecidas.

Figura 65

Escala de valores de SUS



Nota. Adaptado de (Userlytics, 2017).

Con el objetivo de utilizar una escala de valores de SUS, dentro del cuestionario CSUQ, se realizó mediante la siguiente fórmula, presentada por Lewis (2018), donde menciona que P representa cada pregunta dentro del cuestionario.

$$CSUQ = 100 - \left(\frac{\sum_{n=1}^{16} P_n}{16} - 1 \right) * \frac{100}{6}$$

Esta fórmula debe ser utilizada por cada encuestado.

3.2.3. Análisis general de los resultados de la encuesta

A continuación, en la **Tabla 28**, se muestran los resultados obtenidos de todas las preguntas hechas a los encuestados.

Tabla 28

Tabla de frecuencia

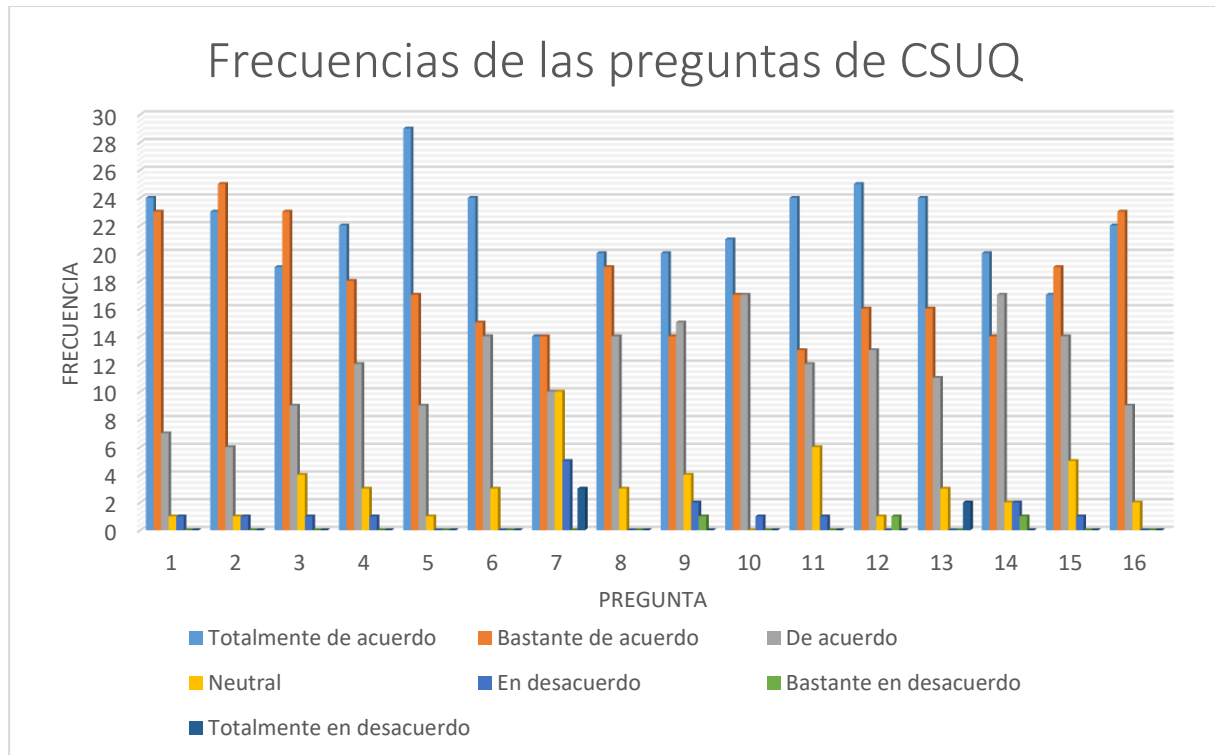
OPCIONES	PREGUNTAS															
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
Totalmente de acuerdo Bastante de acuerdo	24	23	19	22	29	24	14	20	20	21	24	25	24	20	17	22
De acuerdo Neutral En desacuerdo Bastante en desacuerdo Totalmente en desacuerdo	23	25	23	18	17	15	14	19	14	17	13	16	16	14	19	23
	7	6	9	12	9	14	10	14	15	17	12	13	11	17	14	9
	1	1	4	3	1	3	10	3	4	0	6	1	3	2	5	2
	1	1	1	1	0	0	5	0	2	1	1	0	0	2	1	0
	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0
	0	0	0	0	0	0	3	0	0	0	0	0	2	0	0	0
TOTAL	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56

Nota. Autoría propia.

Se puede observar que la respuesta que tiene mayor elección fue la de “Totalmente de acuerdo”, con un total de 348 puntos. En la **Figura 66**, se representó el diagrama de barras que corresponde a la **Tabla 28**.

Figura 66

Diagrama de frecuencias del cuestionario CSUQ

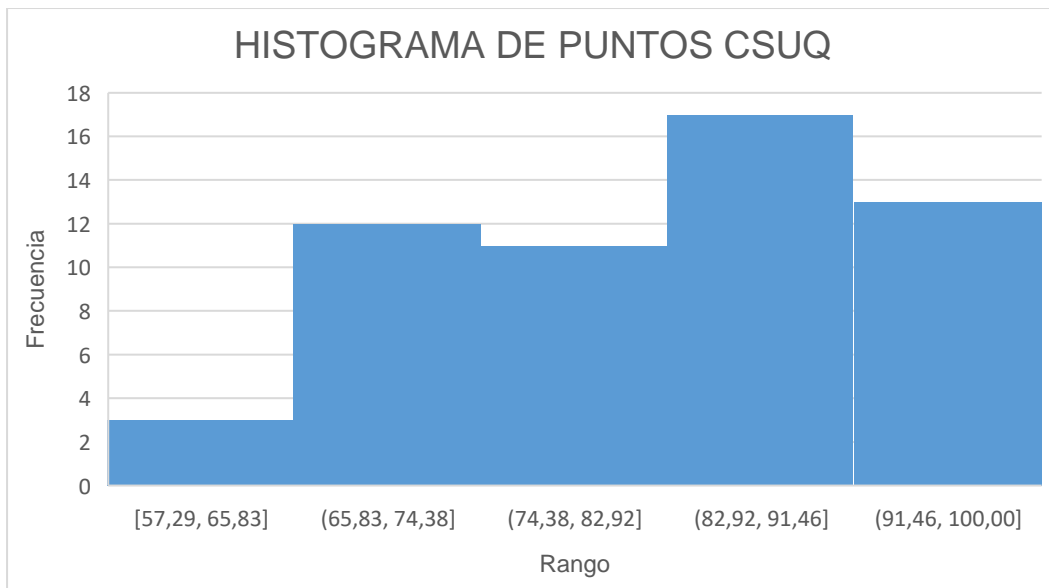


Nota. Autoría propia.

Para completar el análisis que brinda CSUQ, con respecto a la satisfacción, se realizó un histograma, que se muestra en la **Figura 67**. Los resultados evidencian que el rango con mayor frecuencia es de [82.92 – 91.46], lo que indica que el sistema se encuentra entre las categorías de “Excelente” y ‘Lo Mejor Imaginable’ en la escala SUS, con un grado B según la **Figura 65**. Esto sugiere una alta satisfacción, aunque el puntaje aún no alcanza el rango máximo de “Lo Mejor Imaginable”

Figura 67

Histograma de los puntajes CSUQ



Nota. Autoría propia.

Además, se calculó el promedio de los puntajes CSUQ, obteniendo un valor de **82,89**, lo cual, según la **Figura 65**, corresponde a "Excelente" con un grado B. Esta es la calificación asignada al sistema desarrollado.

3.3. Proceso de Evaluación del Modelo de Calidad en Uso

Se realizó la tabulación de los datos obtenidos de la encuesta para obtener los resultados, y luego se aplicaron las fórmulas matemáticas con las métricas que se presentan a continuación:

Característica de satisfacción y sub característica de utilidad

3.3.1. Satisfacción del usuario

Esta métrica evalúa qué tan satisfechos están los usuarios con el software. Se realiza mediante un cuestionario que recoge las opiniones de los usuarios sobre su experiencia con el sistema.

Un alto nivel de satisfacción indica que el software cumple con las expectativas de los usuarios y que su experiencia de uso es positiva. Esto puede reflejar una buena usabilidad,

funcionalidad y relevancia del software para las tareas que deben realizar. Los resultados que se pueden obtener están entre 0 en el peor caso y 1 como valor deseado al aplicar la siguiente fórmula:

$$X = \frac{A}{B}$$

En donde:

3.4. A: Cantidad de personas satisfechas

3.5. B: Número total de personas encuestadas

Tabla 29

Resultado de la encuesta CSUQ respecto a la satisfacción

Literal	Suma de pregunta	Máximo valor	Satisfechos
A	344	7	49,1
P	341	7	48,7
C	329	7	47,0
O	322	7	46,0
N	321	7	45,9
M	327	7	46,7
Total			283,4 → 283.4/7
Satisfechos			40

Nota. Autoría propia.

Una vez se obtuvo el valor de usuarios satisfechos respecto a las preguntas relacionadas con la métrica de satisfacción clasificadas del cuestionario CSUQ, se procedió a calcular el promedio total de usuarios satisfechos, para ello se realizó a sumatoria de la columna Satisfechos, y se dividió para el número de preguntas de dicha categoría.

Ahora para el cálculo final de usuarios satisfecho, respecto a todos los usuarios, se aplica la fórmula descrita anteriormente $X=A/B$, donde X es el promedio de usuarios satisfechos, y B es el número total de encuestados. Los valores son presentados en la **Tabla 30**.

Tabla 30

Métrica de satisfacción

Elemento	Detalle	Valor
A	Número de usuarios satisfechos	40
B	Número total de usuarios	56

Nota. Autoría propia.

Reemplazando las variables por los de la tabla dio como resultado:

$$X = \frac{40}{56} = 0.71$$

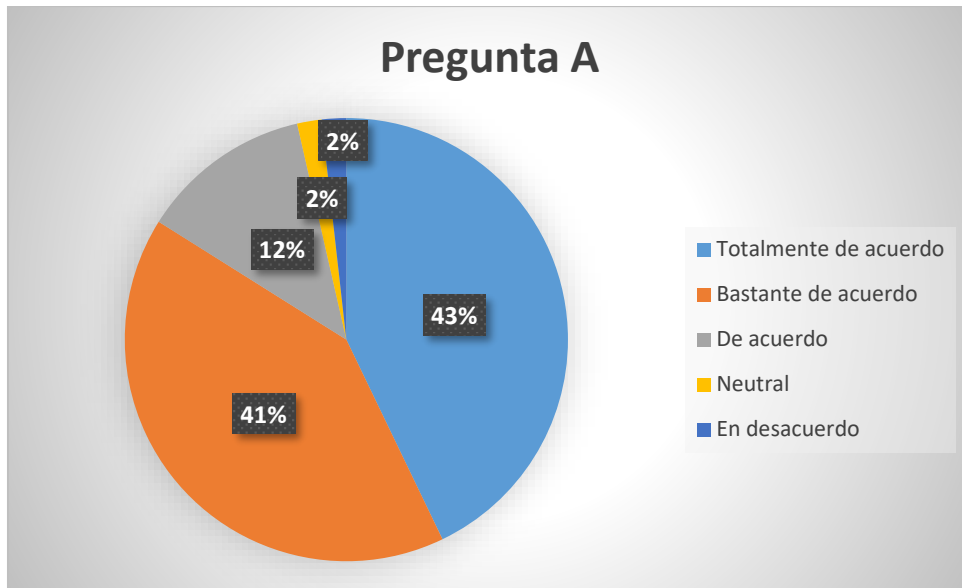
El análisis de los resultados muestra que el sistema tiene un nivel alto de satisfacción, alcanzando un 71% de usuarios satisfechos. Esto refleja que el software está cumpliendo con las expectativas de la mayoría de los usuarios, destacando aspectos como la usabilidad y la funcionalidad. El valor obtenido es cercano a 1, lo que indica que el sistema tiene un impacto positivo en la experiencia del usuario.

A continuación, se presentan los respectivos análisis para la característica de Satisfacción de la norma ISO/IEC 25010, enfocándose en la subcaracterística de Utilidad y utilizando en este apartado la métrica de satisfacción.

Pregunta A: En general, estoy satisfecho con lo fácil que es usar este sistema.

Figura 68

Gráfico de pastel de la pregunta A



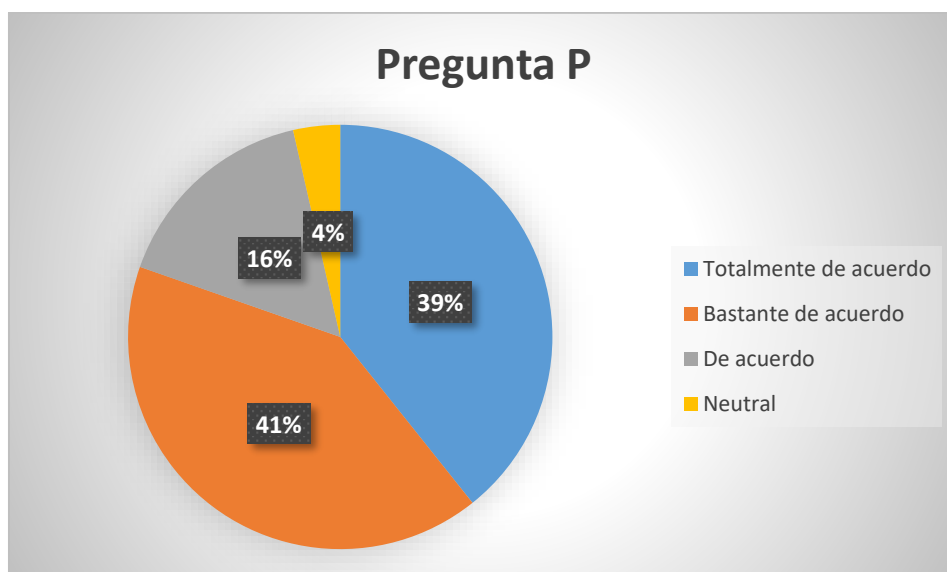
Nota. Autoría propia.

El 96% de los encuestados manifestaron estar satisfechos con lo fácil que les resultó usar la aplicación, desglosando esto es un 43%, 41% y 12% totalmente de acuerdo, bastante de acuerdo y de acuerdo, respectivamente. Al tener estas tres respuestas como las más elegidas, denota una satisfacción alta con respecto a su uso.

Pregunta P: En general, estoy satisfecho con este sistema.

Figura 69

Gráfico de pastel de la pregunta P



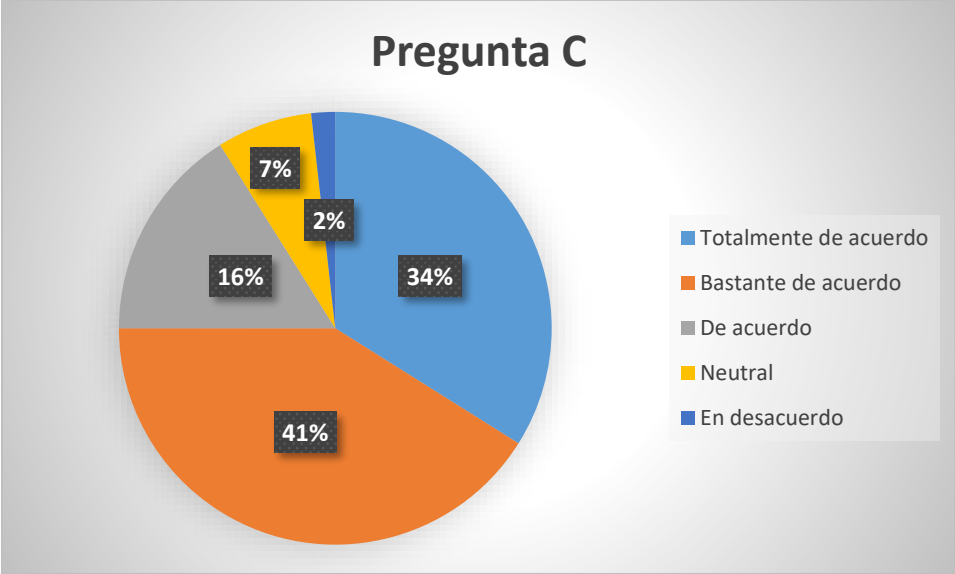
Nota. Autoría propia.

El 96% de los encuestados afirman positivamente su satisfacción respecto al sistema, desglosando como 39%, 41% y 16%, totalmente de acuerdo, bastante de acuerdo y de acuerdo, respectivamente. Al mantener un porcentaje alto de respuestas positivas, afirma que casi la totalidad de encuestados estuvieron satisfechos con el uso general de la aplicación.

Pregunta C: Pude completar mi trabajo rápidamente usando este sistema.

Figura 70

Gráfico de pastel de la pregunta C



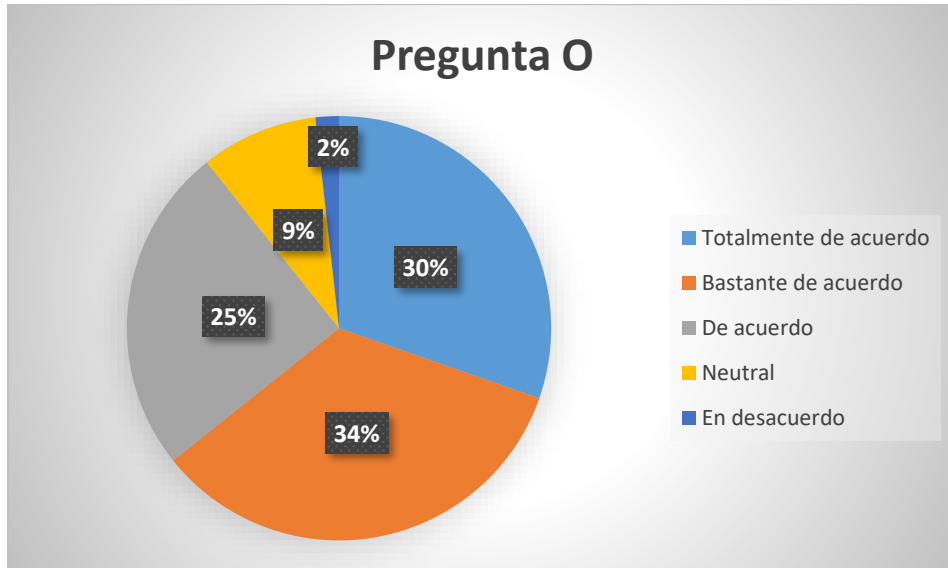
Nota. Autoría propia.

El 91% de los encuestados expresan respuestas positivas respecto a poder completar su trabajo rápidamente usando la aplicación, esto con respecto a la funcionalidad de la identificación y muestra de información en tiempo real. Aquí también se observa que un 7% de los encuestados se mantienen en una posición neutral respecto a la culminación del objetivo del sistema.

Pregunta O: Este sistema tiene todas las funciones y capacidades que espero que tenga.

Figura 71

Gráfico de pastel de la pregunta O



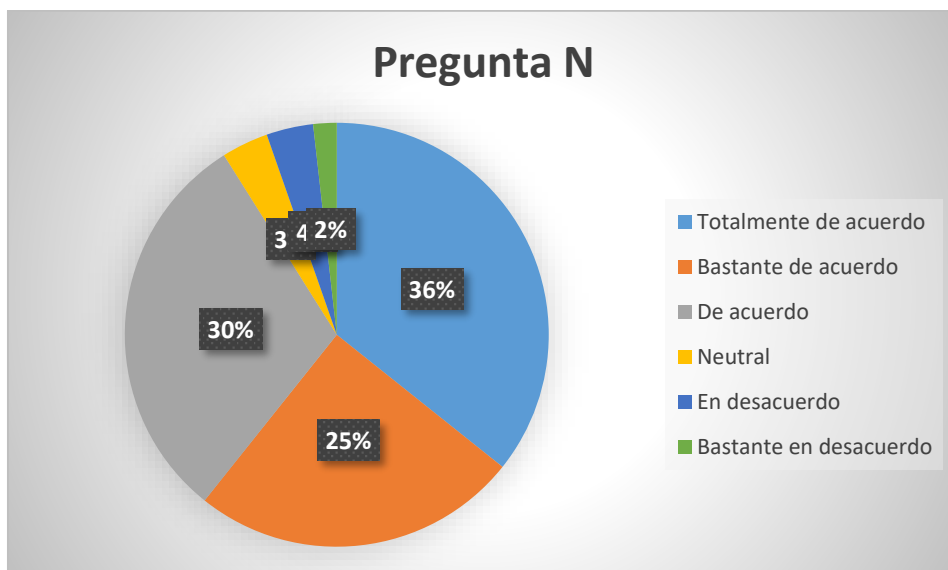
Nota. Autoría propia.

El 89% de los encuestados indicaron una respuesta positiva respecto a las funciones y capacidades de la aplicación, siendo un porcentaje alto, lo cual demostró que en su gran mayoría los encuestados tuvieron éxito al identificarse y visualizarse dentro de la aplicación.

Pregunta N: Me gustó usar la interfaz de este sistema.

Figura 72

Gráfico de pastel de la pregunta N



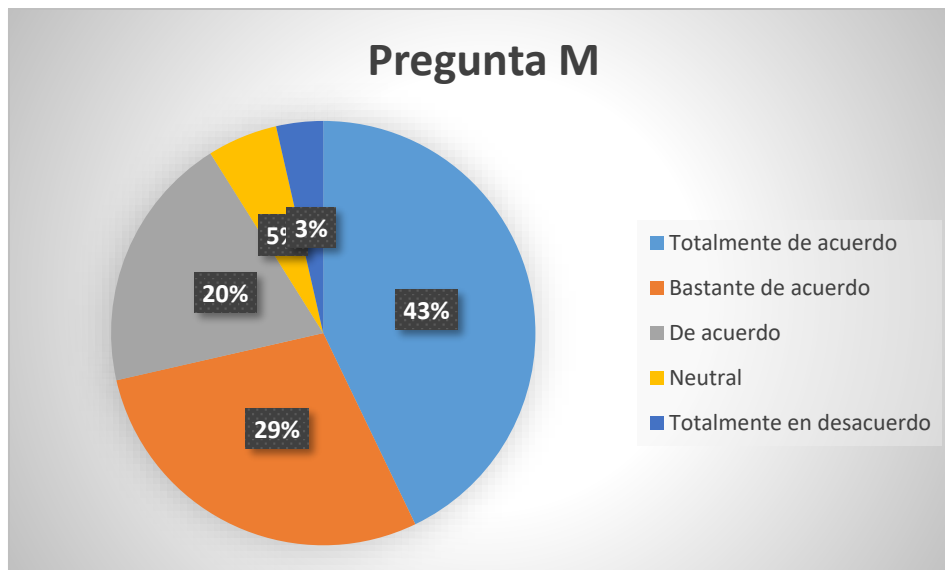
Nota. Autoría propia.

El 91% de los encuestados expresan que les gustó la interfaz presentada en el sistema, lo cual significó que además de la funcionalidad esperada, usarían el sistema por el gusto que tuvieron ante la interfaz del mismo.

Pregunta M: La interfaz de este sistema fue agradable.

Figura 73

Gráfico de pastel de la pregunta M



Nota. Autoría propia.

El 91% de los encuestados expresan que la interfaz presentada en el sistema es de su agrado, lo cual significó que una interfaz sencilla y minimalista es mayormente popular en estudiantes. Con una pantalla de carga clara, y una información concisa que se espera en dicha aplicación.

3.3.2. Métrica de confianza

Determina el nivel de confianza de los usuarios que utilizaron la aplicación, considerando las experiencias de insatisfacción ocasionadas por fallos en el sistema. Para calcularlo, se emplean las siguientes fórmulas:

$$X = \frac{A}{B}$$

$$C = 1 - X$$

En donde:

A: Número de quejas percibidas

B: Número total de usuarios

X: Proporción de usuarios insatisfechos

C: Nivel de confianza de los usuarios

Tabla 31

Métrica de confianza

Elemento	Detalle	Valor
A	Número de quejas percibidas en las preguntas [G – L]	14
B	Número total de usuarios	56

Nota. Autoría propia.

Reemplazando en la fórmula mencionada anteriormente para calcular la proporción de usuarios insatisfechos, se obtiene lo siguiente:

$$X = \frac{14}{56} = 0.25$$

Y para calcular el nivel de confianza que se obtiene con respecto a las quejas percibidas, se emplea la siguiente fórmula dicha previamente, dando como resultado lo siguiente:

$$C = 1 - 0.25 = 0.75$$

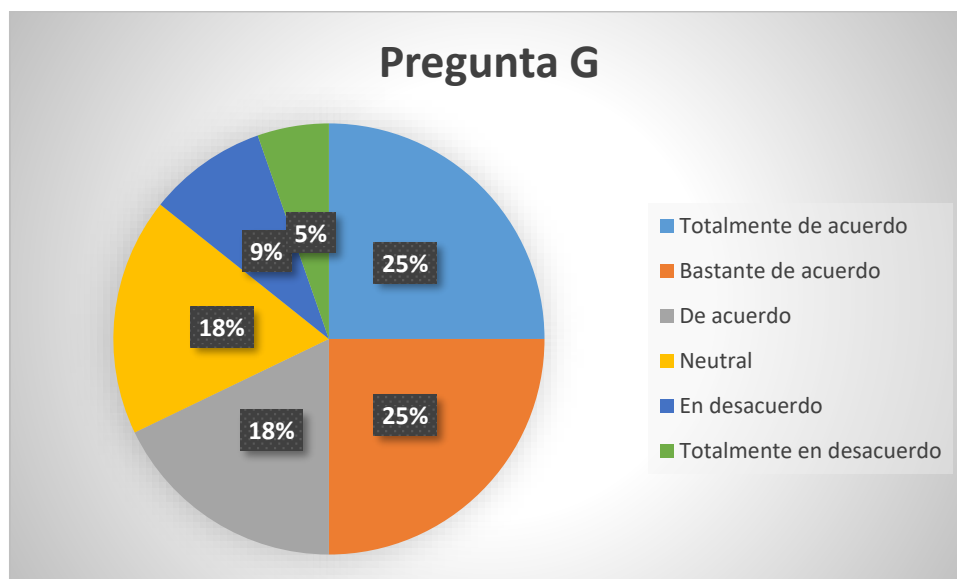
El nivel de confianza alcanzado, del 75%, demuestra que la mayoría de los usuarios confía en el funcionamiento de la aplicación, lo que refleja una percepción general favorable hacia su desempeño. Este resultado evidencia que tres de cada cuatro personas han tenido una experiencia satisfactoria, destacando la solidez del sistema y su capacidad para cumplir con las expectativas de los usuarios en su mayoría. Este logro es un indicador alentador del buen camino en el desarrollo de la aplicación.

A continuación, se presentan los respectivos análisis para la característica de Satisfacción de la norma ISO/IEC 25010, enfocándose en la subcaracterística de Utilidad y utilizando en este apartado la métrica de confianza.

Pregunta G: El sistema dio mensajes de error que me indicaron claramente cómo solucionar problemas.

Figura 74

Gráfico de pastel de la pregunta G



Nota. Autoría propia.

El 50% de los encuestados afirmó estar totalmente de acuerdo y bastante de acuerdo con que el sistema proporcionó mensajes de error claros para solucionar problemas, lo cual es un resultado muy positivo, ya que refleja que la mitad de los usuarios encontró útil y explícita esta funcionalidad.

Por otro lado, el 18% señaló estar neutral, lo que sugiere que estos usuarios no tuvieron problemas significativos o simplemente no prestaron mucha atención a los mensajes de error.

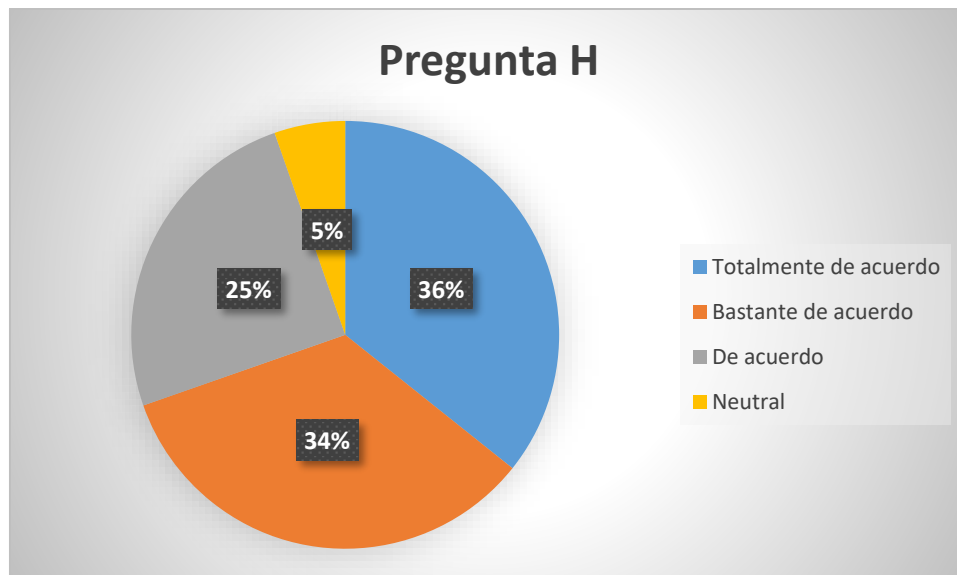
Sin embargo, un 14% expresó opiniones negativas (en desacuerdo y totalmente en desacuerdo), lo que indica que, para este grupo, los mensajes de error no fueron lo suficientemente claros o efectivos. Aunque las respuestas positivas predominan, estos

resultados muestran la necesidad de mejorar aún más los mensajes de error para asegurar que sean comprensibles y útiles para todos los usuarios.

Pregunta H: Cada vez que cometía un error al utilizar el sistema, podía recuperarme fácil y rápidamente.

Figura 75

Gráfico de pastel de la pregunta H



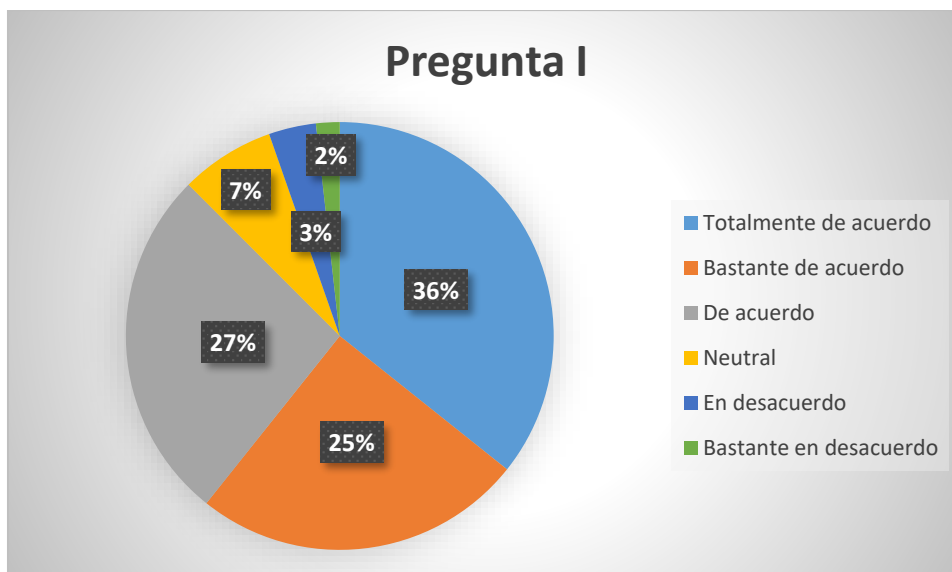
Nota. Autoría propia.

El 95% de los encuestados afirma que, al obtener un error dentro de la aplicación, pudieron continuar con el servicio, casi al instante, estando totalmente de acuerdo, bastante de acuerdo y de acuerdo. Lo que denota una solvencia ante errores alta.

Pregunta I: La información (como ayuda en línea, mensajes en pantalla y otra documentación) provista con este sistema era clara.

Figura 76

Gráfico de pastel de la pregunta I



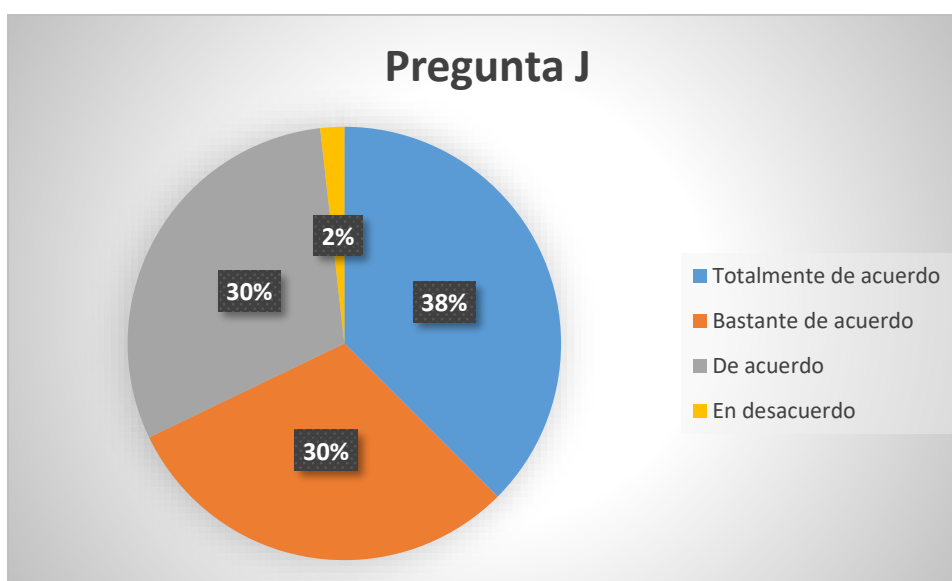
Nota. Autoría propia.

El 88% de los encuestados están conformes con la información presentada en pantalla (información del rostro identificado) siendo clara con lo que se mostraba en tiempo real. Pese a tener en esta pregunta un 5% de respuestas negativas, el mayor porcentaje se encuentra liderado por respuestas positivas.

Pregunta J: Fue fácil encontrar la información que necesitaba.

Figura 77

Gráfico de pastel de la pregunta J



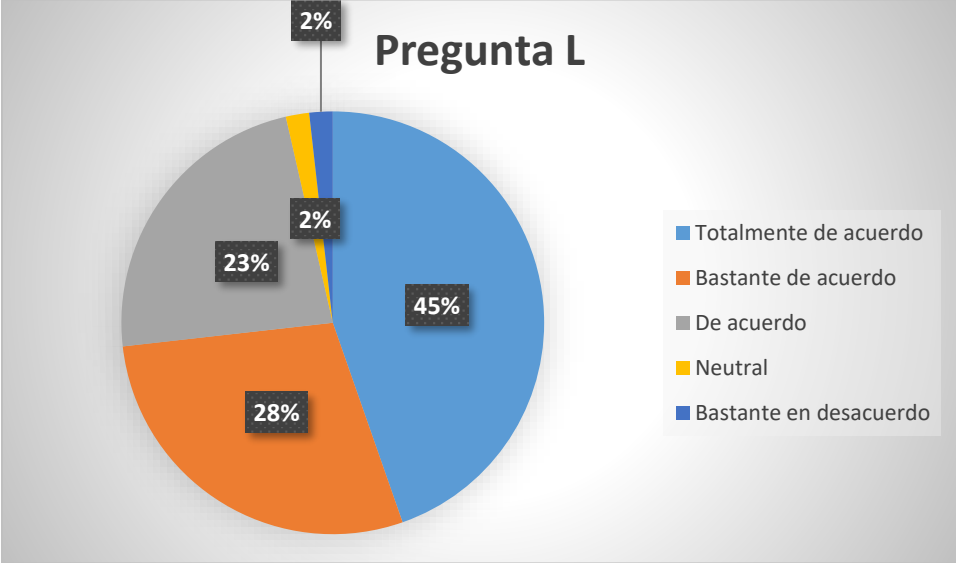
Nota. Autoría propia.

El 98% de los encuestados aseguran de manera positiva que la información que se esperaba fue fácil de encontrar, debido a la presentación central de la pantalla, que es donde se mostró la información personal de las personas registradas. Esto demostró un gran éxito con la función que se esperaba de la realidad aumentada.

Pregunta L: La organización de la información en las pantallas del sistema fue clara.

Figura 78

Gráfico de pastel de la pregunta L



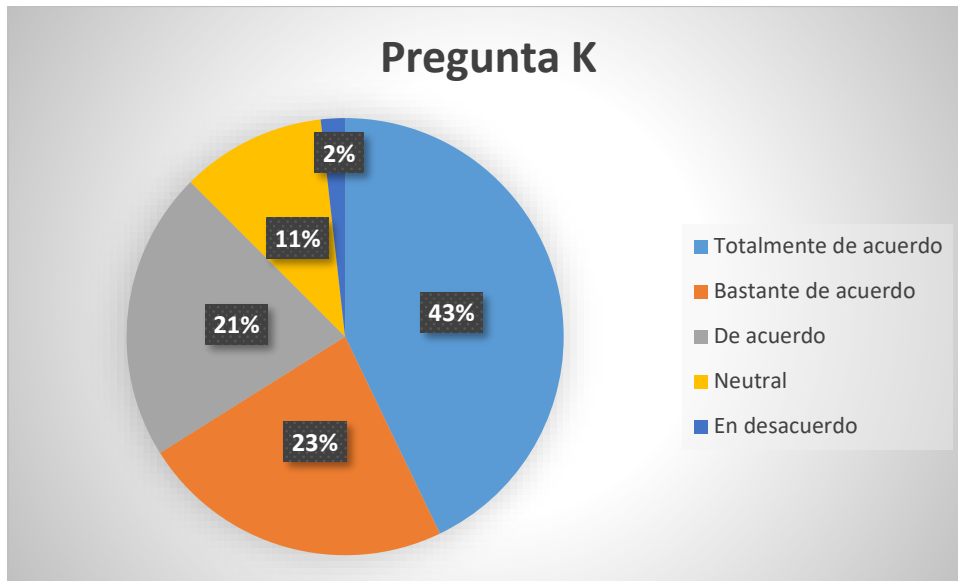
Nota. Autoría propia.

El 96% de los encuestados afirman que la posición y la forma en la que se mostró la información con realidad aumentada, en tiempo real, fue clara.

Pregunta K: La información provista por el sistema fue efectiva para ayudarme a completar mi trabajo.

Figura 79

Gráfico de pastel de la pregunta K



Nota. Autoría propia.

El 87% de los encuestados mencionaron que los datos presentados con realidad aumentada, dentro de la aplicación, eran los necesarios, que junto a una rápida identificación facial, les facilitó el reconocimiento, simulando un entorno agresivo, o pasivo.

3.3.3. Métrica de comodidad

Aquí se describe que tan fácil les resultó a los usuarios utilizar la aplicación móvil, junto con la cantidad de esfuerzo que tuvieron que hacer para interactuar con la misma. Esta métrica se calcula ponderando los valores de las respuestas a cada pregunta, según la escala correspondiente. El valor final se obtiene sumando las respuestas ponderadas y dividiendo el total entre el número de usuarios, según la fórmula:

$$X = A + B + C + D + E + F + G$$

En donde:

A – B – C – D – F – G: Valor calculado de las respectivas preguntas con relación a la comodidad.

Tabla 32*Métrica de comodidad*

Elemento	Detalle	Ponderación	Total Respuestas	Valor	Valor Final
A	Totalmente de acuerdo	1	98	98	0,438
B	Bastante de acuerdo	0,84	75	63	0,281
C	De acuerdo	0,7	41	28,7	0,128
D	Neutral	0,56	8	4,48	0,020
E	En desacuerdo	0,42	2	0,84	0,004
F	Bastante en desacuerdo	0,28	0	0	0,000
G	Totalmente en desacuerdo	0,14	0	0	0,000

Nota. Autoría propia.

Al reemplaza los valores en la formula mencionada previamente se obtiene lo siguiente:

$$X = 0.438 + 0.281 + 0.128 + 0.020 + 0.004 + 0 + 0 = 0.87$$

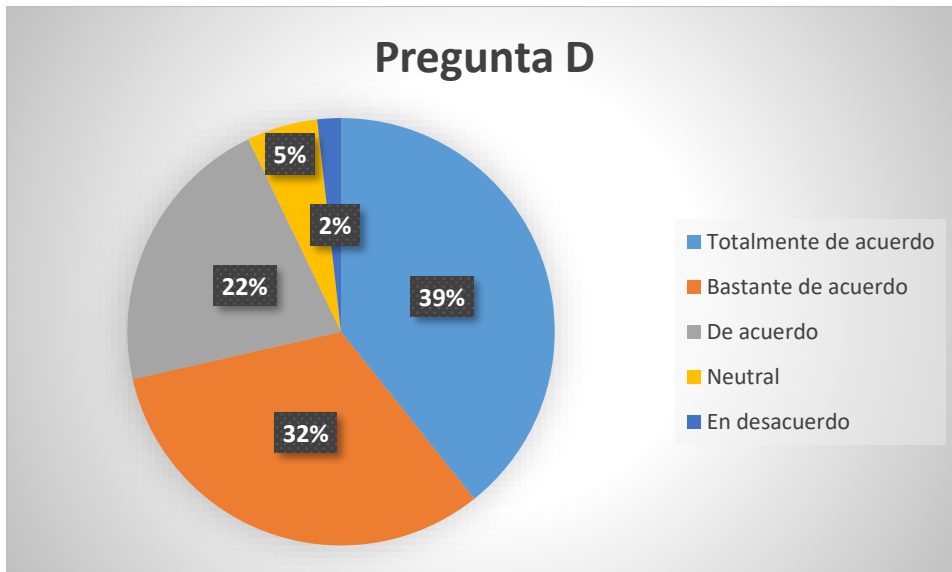
El valor de 0.87 en la métrica de comodidad indica que la mayoría de los usuarios encontró la aplicación fácil de usar, con poco esfuerzo necesario para interactuar con ella. Este alto valor refleja una experiencia de usuario positiva, ya que las respuestas más frecuentes fueron de acuerdo con la facilidad de uso.

A continuación, se presentan los respectivos análisis para la característica de Satisfacción de la norma ISO/IEC 25010, enfocándose en la subcaracterística de Utilidad y utilizando en este apartado la métrica de comodidad.

Pregunta D: Me sentí cómodo usando este sistema.

Figura 80

Gráfico de pastel de la pregunta D



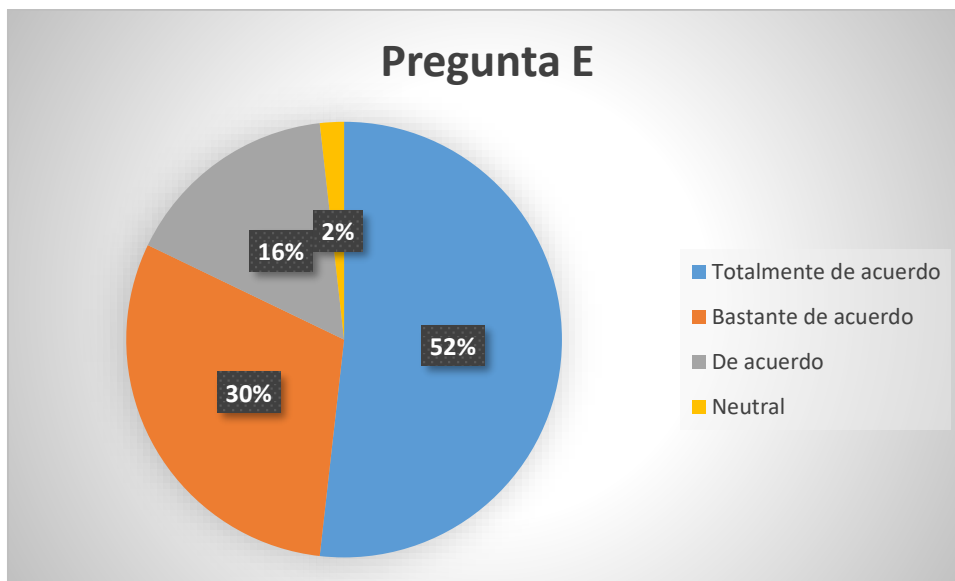
Nota. Autoría propia.

El 93% de los encuestados respondieron de manera positiva ante la comodidad que les representaba el usar la aplicación, hubo un 5% que se mantuvo neutral, dado a diversos factores, como conexión a internet, donde un 2% igualmente por los mismos factores antes mencionados, respondieron de manera negativa.

Pregunta E: Fue fácil aprender a usar este sistema.

Figura 81

Gráfico de pastel de la pregunta E



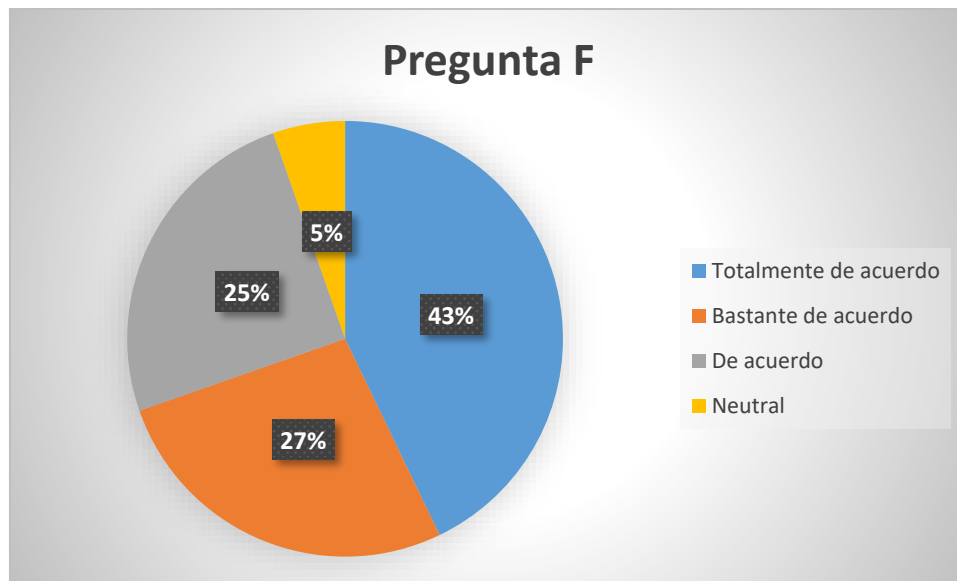
Nota. Autoría propia.

El 98% de los usuarios afirman de manera positiva que el sistema no le dio mayores complicaciones a la hora de saber cómo ejecutarlo, esto demostró que la aplicación tuvo éxito con respecto a una interfaz amigable, y entendible.

Pregunta F: Creo que podría ser productivo rápidamente usando este sistema.

Figura 82

Gráfico de pastel de la pregunta F



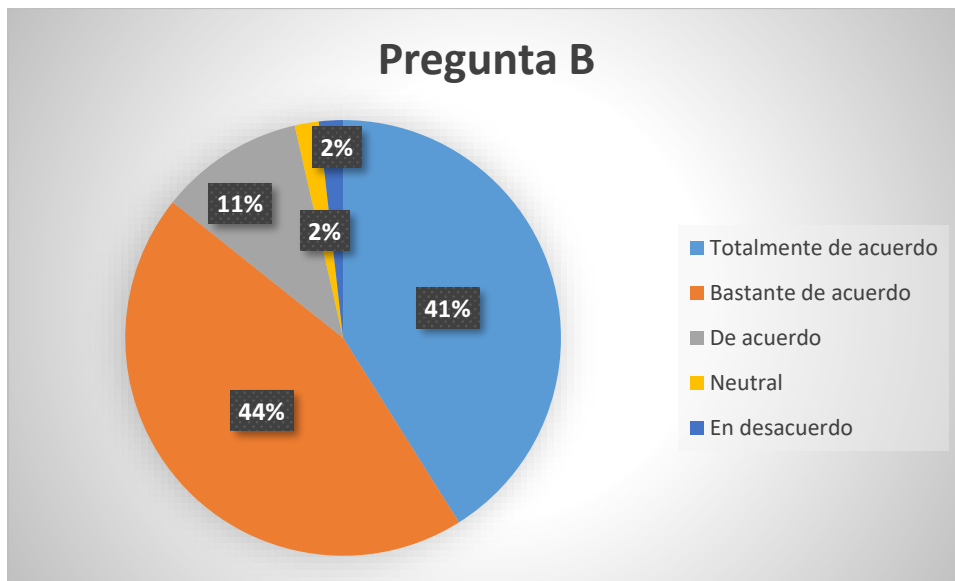
Nota. Autoría propia.

El 95% de los encuestados afirman que con respecto a la función que tiene el sistema, podrían ser más productivos en situaciones donde podrían o deberían de usar la aplicación.

Pregunta B: Fue sencillo usar este sistema.

Figura 83

Gráfico de pastel de la pregunta B



Nota. Autoría propia.

El 95% de los encuestados respondieron de forma positiva con respecto a la facilidad que se tiene a la hora de usar el sistema, se puede contrastar con el 2% de usuarios Neutrales, y 2% de Usuarios en Desacuerdo, que optaron por responder de manera negativa el cómo perciben lo fácil que es usar la aplicación.

3.4. Resultado de la evaluación

Después de llevar a cabo la validación de calidad de la aplicación móvil, los resultados obtenidos son los siguientes. En la **Tabla 33** se presenta un resumen con el peso y los valores correspondientes a cada métrica. Además, en la **Figura 84** se muestra un resumen con los valores finales de la evaluación de calidad realizada a la aplicación.

Tabla 33*Resultado de la evaluación*

Característica	Subcaracterística	Métrica	Peso (%)	Medición	Resultado	Total
		Satisfacción	25	0.71	17.75	
Satisfacción	Utilidad	Confianza	35	0.75	26.25	78.8
		Comodidad	40	0.87	34.8	

Nota. Autoría propia.**Figura 84***Resultados de evaluación de calidad uso*

RESULTADOS DE EVALUACIÓN DE CALIDAD INTERNA, EXTERNA Y EN USO							
	Características	Valor Parcial Total (/10)	Nivel de Importancia	Porcentaje de Importancia	Valor Final	Calidad Parcial del Sistema (/10)	Calidad Total del Sistema (/10)
CALIDAD EN USO	Efectividad	0,00	0	0%	0,00	7,88	7,88
	Eficiencia	0,00	0	0%	0,00		
	Satisfacción	7,88	0	100%	7,88		
	Libertad de Riesgo	0,00	No Aplica	0%	0,00		
	Cobertura de contexto	0,00	0	0%	0,00		

Nota. Autoría propia.

Para interpretar los resultados obtenidos, se utilizó como base la escala definida por la norma ISO/IEC 25040, que establece cómo gestionar y supervisar la evaluación del software desde una perspectiva más amplia. Esta norma parte de la definición del propósito de la evaluación, aplicando recomendaciones, requisitos y orientaciones tanto para programadores como para consumidores y evaluadores. Además, se encarga de verificar la gestión intermedia y final de los procesos relacionados con la información recolectada (Paola et al., 2019). Por lo tanto, en la **Figura 85** se presenta un rango de puntuaciones que permite clasificar la aplicación y determinar si existen falencias o no.

Figura 85

Niveles de puntuación

Valor de medición	Nivel de puntuación	Grado de satisfacción
7.91-10	Cumple con los requisitos	Muy satisfactorio
4.91-7.9	Aceptable	Satisfactorio
1.91-4.9	Mínimamente aceptable	No satisfactorio
0-1.9	Inaceptable	No satisfactorio

Nota. (Paola et al., 2019).

Luego de finalizar el proceso de validación de los datos obtenidos mediante la encuesta CSUQ, se puede afirmar que la aplicación móvil desarrollada cumple con los criterios establecidos en la norma ISO/IEC 25010. Como se observa en los resultados obtenidos, la aplicación destinada al reconocimiento facial con realidad aumentada alcanzó una puntuación de 7.88.

Este puntaje la sitúa dentro del rango de “Aceptable” y con un nivel de “Satisfactorio”, lo que sugiere que la aplicación es intuitiva, fácil de utilizar y no requiere conocimientos técnicos avanzados para su manejo adecuado.

3.5. Test de Normalidad

Se aplicó el test de normalidad a los resultados de cada pregunta para determinar si siguen una distribución normal o no.

Se aplicó el test de Kolmogorov a los resultados de cada pregunta, ya que la muestra supera los 50 participantes. Para ello, se utilizó el software estadístico SPSS y Python, con el fin de obtener una doble validación del test.

Los resultados se presentan en la **Tabla 34**. En el análisis de este test, se estableció un nivel de significancia de 0,05. Esto significa que si el p-value es menor o igual a 0,05, los datos no siguen una distribución normal; de lo contrario, si el p-value es mayor a 0,05, los datos tienen una distribución normal, según lo indicado por (Minitab, 2019).

Tabla 34*Resultado del test de Kolmogorov*

Preguntas	Valor de distribución
1	0.0005
2	0.0001
3	0.0003
4	0.0089
5	0.0005
6	0.0016
7	0.0314
8	0.0150
9	0.0340
10	0.0105
11	0.0017
12	0.0021
13	0.0026
14	0.0243
15	0.0114
16	0.0014

Nota. Autoría propia.

3.6. Correlación de Variables

Se realizaron pruebas específicas para analizar la correlación de variables, empleando el método de Spearman, dada que las variables no son normales, por lo cual se decidió emplear el test antes dicho, para pruebas no paramétricas.

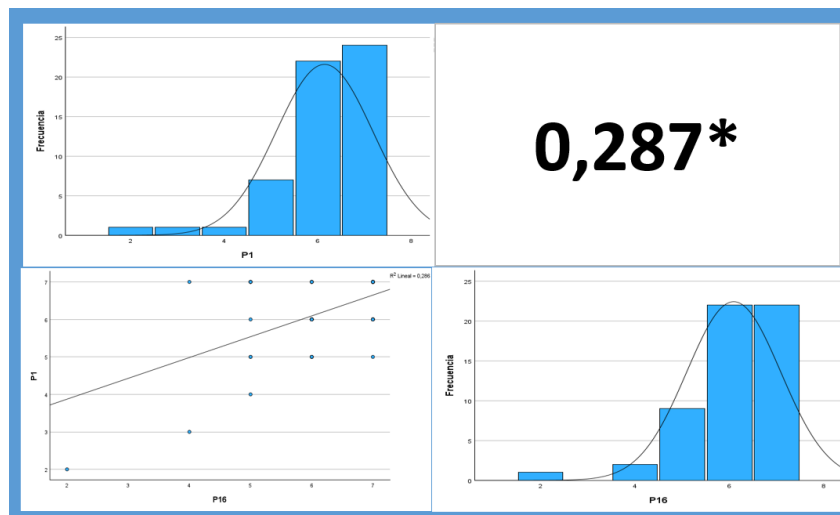
Cada una de estas pruebas está vinculada con la Característica de **Satisfacción** y subcaracterística de **Utilidad**, según lo establece la norma ISO/IEC 25010, y con las preguntas correspondientes de la encuesta CSUQ.

A continuación, se presenta el análisis obtenido mediante el software SPSS Statistics, y verificado con Python, de las preguntas relacionadas, junto con los resultados derivados de los métodos mencionados previamente.

La pregunta 1 ("En general, estoy satisfecho con lo fácil que es usar este sistema") y la pregunta 16 ("En general, estoy satisfecho con este sistema") muestran una correlación positiva y significativa, aunque débil. Esto indica que la facilidad de uso tiene alguna relación con la satisfacción general del sistema, pero no es el único factor que influye en la percepción global. Puede haber otros elementos, como las funciones ofrecidas o el diseño, que afectan la satisfacción general.

Figura 86

Correlación entre las preguntas 1 y 16

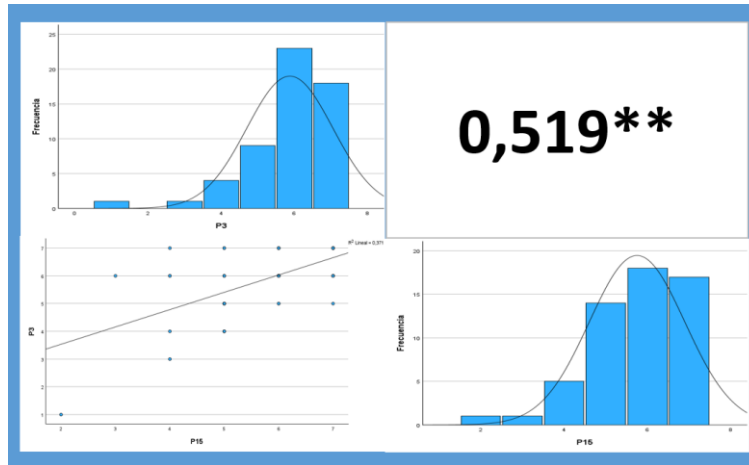


Nota. Autoría propia.

La pregunta 3 ("Pude completar mi trabajo rápidamente usando este sistema") y la pregunta 15 ("Este sistema tiene todas las funciones y capacidades que espero que tenga") presentan una correlación moderada y significativa. Esto indica que los usuarios que perciben que el sistema es rápido para completar tareas también tienden a considerar que tiene las funcionalidades necesarias. Esto sugiere que las capacidades y funcionalidades del sistema influyen directamente en la eficiencia percibida, y mejorar las herramientas o funciones clave podría aumentar la velocidad de trabajo.

Figura 87

Correlación entre las preguntas 3 y 15

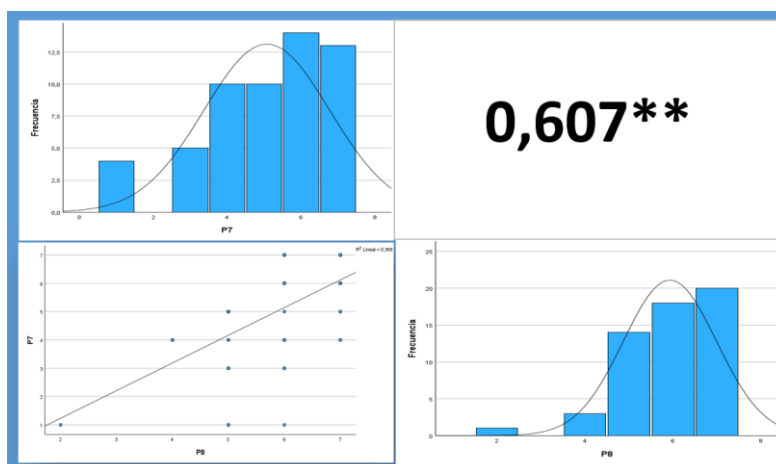


Nota. Autoría propia.

Las preguntas 7 ("El sistema dio mensajes de error que me indicaron claramente cómo solucionar problemas") y 8 ("Cada vez que cometía un error al utilizar el sistema, podía recuperarme fácil y rápidamente") tienen una correlación fuerte y muy significativa. Esto implica que los usuarios que encontraron útiles los mensajes de error también percibieron que podían recuperarse rápidamente de los problemas. Esto destaca la importancia de diseñar mensajes de error claros y útiles para mejorar la experiencia del usuario.

Figura 88

Correlación entre las preguntas 7 y 8

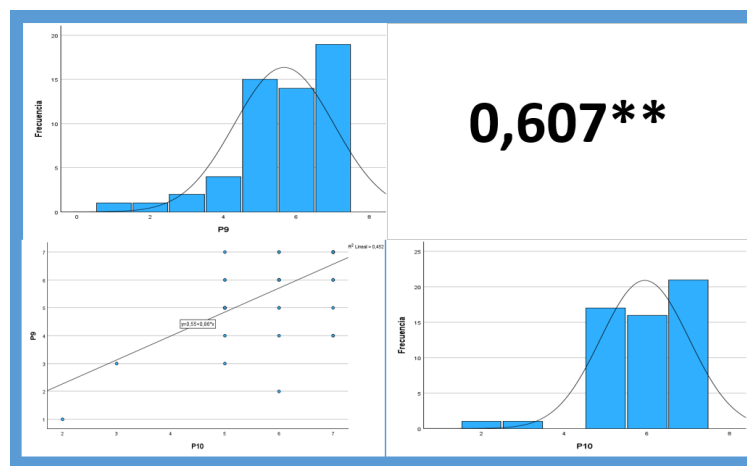


Nota. Autoría propia.

Las preguntas 9 ("La información provista con este sistema era clara") y 10 ("Fue fácil encontrar la información que necesitaba") muestran una correlación fuerte y muy significativa. Esto indica que la claridad de la información proporcionada por el sistema está directamente relacionada con la facilidad para encontrarla. Los usuarios que consideran clara la información también tienden a encontrarla rápidamente, lo que sugiere que una buena organización y presentación de la información son cruciales.

Figura 89

Correlación entre las preguntas 9 y 10

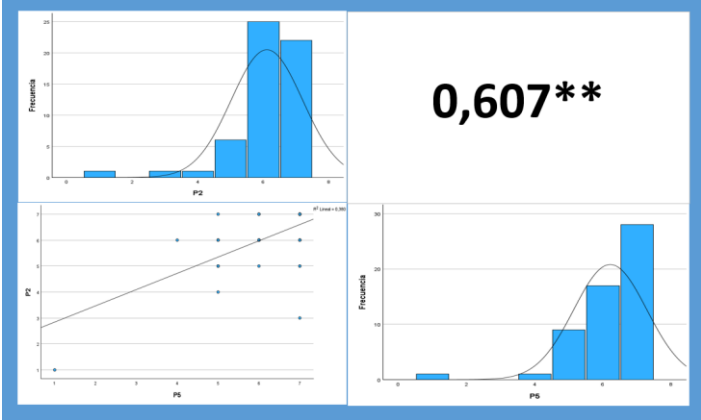


Nota. Autoría propia.

La pregunta 2 ("Fue sencillo usar este sistema") y la pregunta 5 ("Creo que podría ser productivo rápidamente usando este sistema") tienen una correlación moderada y significativa. Esto sugiere que, aunque la facilidad de uso contribuye a la percepción de productividad, no siempre están directamente vinculadas. Es posible que otros factores, como las funcionalidades o el diseño, también jueguen un papel en la productividad percibida.

Figura 90

Correlación entre las preguntas 2 y 5



Nota. Autoría propia.

CONCLUSIONES

La elaboración del marco teórico sobre el desarrollo de aplicaciones móviles con realidad aumentada, junto con su administrador web, garantizó la dirección y viabilidad del proyecto de integración curricular. Este proceso permitió analizar en profundidad las implicaciones tecnológicas y funcionales de desarrollar una aplicación con estas características. Como resultado se identificaron limitaciones en las herramientas inicialmente consideradas, lo que condujo a una decisión estratégica de adoptar AR Foundation y Unity, en lugar de Vuforia y Flutter. Esto se debió a su capacidad multiplataforma e integración con la herramienta de realidad aumentada. Ayudó a eliminar la redundancia de compilar una aplicación de Unity dentro de Flutter. Mediante la redacción se logró comprender la relación general entre herramientas y la importancia que significa cada una al desarrollo.

La creación de la aplicación móvil y su administrador web, usando la metodología SCRUM, permitió el control de avances, y situaciones no previstas en un inicio, lo que permitió adaptarse a los retrasos dentro del desarrollo, tanto por errores de programación, como demora de la aceptación del cambio de tecnologías por la incapacidad de relacionarse entre sí, replanificando los Sprints de manera eficiente. La retroalimentación semanal constante ofreció un control en su gran mayoría de lo esperado, de lo que se debía mejorar, y de los resultados que se esperaban continuamente.

La evaluación de la satisfacción de los usuarios, basada en la norma ISO/IEC 25010, arrojó un nivel de satisfacción del **7.88%** en calidad en uso, según los resultados de la encuesta CSUQ. Dado que los datos no eran normales, se aplicó el test de Kolmogorov y el método de Spearman, identificando correlaciones significativas entre variables clave, como la facilidad de uso y la satisfacción general, así como entre funcionalidades y eficiencia percibida. Estos hallazgos destacan que el sistema cumple con los estándares de calidad establecidos, ofreciendo una experiencia útil y satisfactoria que responde a las expectativas de los usuarios y valida la efectividad del proyecto.

RECOMENDACIONES

Considerar una solución diferente para la identificación facial, ya que la implementación de face-api.js requiere el uso de una API externa que puede generar un tiempo de respuesta débil, especialmente al ser consumida dentro de Unity. Explorar otras opciones nativas o bien integradas directamente en Unity podría optimizar el rendimiento y la eficiencia en términos de tiempos de respuesta y procesamiento.

Mantener la metodología SCRUM es altamente recomendable para futuras etapas del proyecto, particularmente al agregar nuevas funcionalidades como por ejemplo la toma de asistencia con reconocimiento facial u alguna otra que se desee emplear. Esta metodología facilita una organización eficiente, permite adaptarse rápidamente a cambios y garantiza una gestión adecuada de los avances mediante retroalimentaciones constantes, lo cual es esencial para cumplir con los objetivos establecidos.

Realizar pruebas exhaustivas en diversos dispositivos, ya que se ha detectado que algunos modelos no permiten la instalación de la aplicación (Redmi, Infinix y Honor). Es importante revisar las posibles causas de este comportamiento, como la versión de Android del dispositivo o la necesidad de actualizaciones del sistema operativo, para asegurar la compatibilidad con una mayor gama de terminales.

Se sugiere estar al tanto de las actualizaciones de AR Foundation, especialmente en relación con el uso de la cámara trasera para la realidad aumentada. Actualmente, solo se puede utilizar la cámara frontal, pero las futuras actualizaciones podrían incluir soporte para la cámara trasera, lo cual mejoraría significativamente la experiencia del usuario y ampliaría las posibilidades de implementación en distintos escenarios.

BIBLIOGRAFÍA

- Adjabi, I., Ouahabi, A., Benzaoui, A., & Taleb-Ahmed, A. (2020). Past, present, and future of face recognition: A review. In *Electronics (Switzerland)* (Vol. 9, Issue 8, pp. 1–53). MDPI AG. <https://doi.org/10.3390/electronics9081188>
- Alaimo, M. (2018). *Introducción a la Agilidad y Scrum*. http://es.wikipedia.org/wiki/Herramienta_CASE
- Araneda, O. (2022, September 26). *¿Cómo funciona Scrum? - Atenos*. <https://atenos.com/agile/como-funciona-scrum/>
- Arguello Fuentes, H. (2011). Sistemas de reconocimiento basados en la imagen facial Recognition systems based on the facial image. In *Revista Avances en Sistemas e Informática* (Vol. 8, Issue 3).
- Arias-Orezano, J. F., Reyna Barreto, B. D., & Mamani-Apaza, G. (2021). Repercusión de arquitectura limpia y la norma ISO/IEC 25010 en la mantenibilidad de aplicativos Android. *TecnoLógicas*, 24(52), e2104. <https://doi.org/10.22430/22565337.2104>
- Arroyo, M. (2023). *▷ Scrum: Dif. entre Prod. Backlog y Sprint Backlog?* <https://metodologiascrum.com/metodologia-scrum-eventos-y-artefactos-diferencia-entre-product-backlog-y-sprint-backlog/>
- Asamblea Nacional. (2018). *Ley Orgánica de Educación Superior*. www.lexis.com.ec
- ASQ. (2023). *What is Software Quality?* <https://asq.org/quality-resources/software-quality>
- Aznarte, J. L., Pardos, M. M., & Lacruz López, J. M. (2022). On the use of facial recognition technologies in university: The uned case. *RIED-Revista Iberoamericana de Educacion a Distancia*, 25(1), 261–277. <https://doi.org/10.5944/ried.25.1.31533>
- Azuma, R. T. (1997). A Survey of Augmented Reality. In *Caudell*. <http://www.cs.unc.edu/~azuma>
- Billinghurst, M., Clark, A., & Lee, G. (2014). A survey of augmented reality. In *Foundations and Trends in Human-Computer Interaction* (Vol. 8, Issues 2–3, pp. 73–272). Now Publishers Inc. <https://doi.org/10.1561/11000000049>
- Britton, J. (2021). *What Is ISO 25010?* PERFORCE. <https://www.perforce.com/blog/qac/what-is-iso-25010>
- Brohm, D., Domurath, N., Glanz-Chanos, V., & Grunert, K. G. (2017). Future trends of augmented reality. In *Augmented reality for food marketers and consumers* (pp. 97–104). Brill | Wageningen Academic. https://doi.org/10.3920/978-90-8686-842-1_6
- Busto, L. (2023, October 31). *Presentación de la realidad aumentada con Unity*. Tokio School. <https://www.tokioschool.com/noticias/realidad-aumentada-con-unity/>

- Cabanillas, S. (2022). *Algoritmos de reconocimiento facial, casos de uso y polémica*. <https://bcnvision.es/blog-vision-artificial/algoritmos-de-reconocimiento-facial-casos-de-uso-y-polemica/>
- Cadena, J., Montaluísa, R., Flores, G., Chancúsig, Juan, & Guaypatin, O. (2017). *Reconocimiento facial con base en imágenes*.
- Camilo, A., Álvarez, J., Andrés, S., Trujillo, L., Pontificia, U., & Seccional Bucaramanga, B. (2021). *Análisis y comparación de las metodologías de SCRUM y según PMI gestión de proyectos*.
- Cárdenas, H., Mesa, F., & Suarez, M. (2018). *Augmented reality (RA): applications and challenges for using it in the classroom*.
- Carmigniani, J., Furht, B., Anisetti, M., Ceravolo, P., Damiani, E., & Ivkovic, M. (2011). Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, 51(1), 341–377. <https://doi.org/10.1007/s11042-010-0660-6>
- Carrera de Software. (2020). *Misión y Visión*. <https://software.utn.edu.ec/informacion/mision-y-vision/>
- Chan, J. (2022). *Facial Recognition Technology and Ethical Issues*.
- ChatGPT. (2024). *Imagen de las características de la realidad aumentada*.
- Cohn, M., & Beck, K. (2008). *User Stories Applied*. Addison-Wesley.
- Deloitte. (2024). *Las 5 ceremonias Scrum: claves para la gestión de procesos*. <https://www2.deloitte.com/es/es/pages/technology/articles/ceremonias-scrum.html>
- Estrada, M., Núñez, J., Saltos, P., & Cunuhay, W. (2021). *Systematic review of the SCRUM methodology for software development*. 7, 434–447. <https://doi.org/10.23857/dc.v7i4.2429>
- Fabila, A., Minami, H., & Izquierdo, M. (2014). *La Escala de Likert en la evaluación docente: acercamiento a sus características y principios metodológicos*.
- Fernández de Terán, A. (2019). *La realidad aumentada en la educación: usabilidad y ejemplos de su aplicación*.
- Gyata, T. (2024, April 1). *Desarrollo de Realidad Aumentada con Unity*. <https://www.gyata.ai/es/augmented-reality/augmented-reality-with-unity>
- Hernández, R. (2010). *ESTUDIO DE TÉCNICAS DE RECONOCIMIENTO FACIAL*.
- Hussain, A., Mkpojiogu, E. O. C., Jama Isse, A., & Mohammed, R. A. (2018). Grab mobile application: A usability evaluation. *AIP Conference Proceedings*, 2016. <https://doi.org/10.1063/1.5055456>

- Ichi. (2020). *face-api.js: una forma de crear un sistema de reconocimiento facial en el navegador*. <https://ichi.pro/es/face-api-js-una-forma-de-crear-un-sistema-de-reconocimiento-facial-en-el-navegador-213256611440215>
- idrd. (2020). *IDRD-Facial-Liveness-Whitepaper-Spanish*.
- ISO 25010. (2021). <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>
- ISO 25010. (2022). <https://iso25000.com/index.php/normas-iso-25000/iso-25010>
- Jakl, A. (2021, August 6). *AR Foundation Fundamentals with Unity (Part 1)*. <https://www.andreasjakl.com/ar-foundation-fundamentals-with-unity-part-1/>
- Jaramillo, C. D. (2021). UTILIZACIÓN DEL SISTEMA DE RECONOCIMIENTO FACIAL PARA PRESERVAR LA SEGURIDAD CIUDADANA Use of the facial recognition system to preserve public safety. *El Criminalista Digital*. <http://revistaseug.ugr.es/index.php/cridi/article/view/20899>
- Jiménez, I., Tutores, S., Martín, R., Titular, C. P., Antonio, S., & Álvarez, C. (2018). *Reconocimiento facial basado en redes neuronales convolucionales*.
- Khabarлак, K. S., & Koriashkina, L. S. (2021). *Fast Facial Landmark Detection and Applications: A Survey*.
- Kortli, Y., Jridi, M., Al Falou, A., & Atri, M. (2020). Face recognition systems: A survey. In *Sensors (Switzerland)* (Vol. 20, Issue 2). MDPI AG. <https://doi.org/10.3390/s20020342>
- Lance, L. (2023, September 27). *Los 5 Eventos o Ceremonias de Scrum: La Guía Definitiva en Profundidad*. <https://thedigitalprojectmanager.com/es/metodologias-gestion-proyectos/ceremonias-scrum-simplificadas/>
- Lewis, J. R. (1995). IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. *International Journal of Human-Computer Interaction*, 7(1), 57–78. <https://doi.org/10.1080/10447319509526110>
- Lewis, J. R. (2018). Measuring Perceived Usability: The CSUQ, SUS, and UMUX. *International Journal of Human-Computer Interaction*, 34(12), 1148–1156. <https://doi.org/10.1080/10447318.2017.1418805>
- Lindsjörn, Y., Sjøberg, D. I. K., Dingsøy, T., Bergersen, G. R., & Dybå, T. (2016). Teamwork quality and project success in software development: A survey of agile development teams. *Journal of Systems and Software*, 122, 274–286. <https://doi.org/10.1016/j.jss.2016.09.028>
- Lorena, M., Rojas, F., María, A., & Pimienta, C. (2022). TECNOLOGÍAS DE RECONOCIMIENTO FACIAL EN COLOMBIA: ANÁLISIS COMPARATIVO EN RELACIÓN CON LA PROTECCIÓN DE DATOS. In *Revista Ius et Praxis, Año* (Vol. 29).

- Mahnič, V., & Hovelja, T. (2012). On using planning poker for estimating user stories. *Journal of Systems and Software*, 85(9), 2086–2095. <https://doi.org/10.1016/j.jss.2012.04.005>
- Mancuzo, G. (2021). 5 Valores de Scrum. <https://blog.comparasoftware.com/como-aplicar-5-valores-de-scrum/>
- Manterola, C., Astudillo, P., Arias, E., & Claros, N. (2013). Revisiones sistemáticas de la literatura. Qué se debe saber acerca de ellas. *Cirugia Espanola*, 91(3), 149–155. <https://doi.org/10.1016/j.ciresp.2011.07.009>
- Mateus, S. P., González, N., & Branch, J. W. (2014). Aplicación de redes neuronales artificiales en entornos virtuales inteligentes. *Informacion Tecnologica*, 25(5), 103–112. <https://doi.org/10.4067/S0718-07642014000500015>
- Maxwell, J. (2012). *Law of Connection: Lesson 10 from The 21 Irrefutable Laws of Leadership*.
- Minitab. (2019). *Interpretar los resultados clave para Prueba de normalidad - Minitab*. <https://support.minitab.com/es-mx/minitab/help-and-how-to/statistics/basic-statistics/how-to/normality-test/interpret-the-results/key-results/>
- Moumane, K., Idri, A., El Aouni, F., Laghnimi, J., Benabdellah, N. C., & Hamal, O. (2024). ISO/IEC 25010-based Quality Evaluation of Three Mobile Applications for Reproductive Health Services in Morocco. *Clinical and Experimental Obstetrics and Gynecology*, 51(4). <https://doi.org/10.31083/j.ceog5104088>
- Muensterer, O. J., Lacher, M., Zoeller, C., Bronstein, M., & Kübler, J. (2014). Google Glass in pediatric surgery: An exploratory study. *International Journal of Surgery*, 12(4), 281–289. <https://doi.org/10.1016/j.ijssu.2014.02.003>
- Mühler, V. (2020). *face-api.js*. <https://justadudewhohacks.github.io/face-api.js/docs/index.html>
- Naciones Unidas. (2019). *La Agenda 2030 y los Objetivos de Desarrollo Sostenible: una oportunidad para América Latina y el Caribe*. www.issuu.com/publicacionescepal/stacks
- Navarro, A., Marañón, E., & Morales, R. (2022). Revisión de los métodos de reconocimiento facial en imágenes RGB-D adquiridas mediante un sensor Kinect. *Revista Cubana de Ciencias Informáticas*, 16(2).
- World Health Organization. (2019). *Coronavirus Disease (COVID-19) Situation Reports*. <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports>
- Ormeño, N. (2019, May 15). *ISO 25010 y el desarrollo de software*. <https://normeno.medium.com/iso-25010-y-el-desarrollo-de-software-112393a4b341>

- Paneiva, B. (2023, November 10). *Control ISO 25010 - Calidad en Uso | Guía y Auditoría | Lumiform*. <https://lumiformapp.com/es/checklists-recursos/certificacion-iso-25010>
- Paola, E., Guaña, R., Gabriela, S., Rosado, P., & Quijosaca, F. (2019). *Evaluación de la calidad en uso de un sistema web/ móvil de control de asistencia a clases de docentes y estudiantes aplicando la norma ISO/IEC 25000 SQuaRe*.
- Perdomo, D., Millán, E., & Gallego, A. (2020). *Use of Augmented Reality, Gamification and M-learning*. <https://www.redalyc.org/journal/4139/413962511045/html/>
- Porter, M., & Heppelmann, J. (2017). *Why Every Organization Needs an Augmented Reality Strategy*. Technology And Analytics. <https://hbr.org/2017/11/why-every-organization-needs-an-augmented-reality-strategy>
- Pressman, R. (2010). *Software Engineering: A Practitioner's Approach*. www.mhhe.com/pressman.
- Productiviza. (2024). *Metodología Scrum: Explicado con Ejemplos Prácticos y Reales*. <https://www.productiviza.com/scrum-ejemplos-practicos-y-reales-explicados/>
- Racheva, Z., Daneva, M., Sikkel, K., Wieringa, R., & Herrmann, A. (2010). Do we know enough about requirements prioritization in agile projects: Insights from a case study. *Proceedings of the 2010 18th IEEE International Requirements Engineering Conference, RE2010*, 147–156. <https://doi.org/10.1109/RE.2010.27>
- Radigan, D. (2021). *Backlog del producto: qué es y cómo crearlo*. <https://www.atlassian.com/es/agile/scrum/backlogs>
- Rebeś, P. (2023). *Software Quality Standards—How and Why We Applied ISO 25010 | Monterail*. <https://www.monterail.com/blog/software-qa-standards-iso-25010>
- Rocha, A. (2020). *Analysis of the Quality in Use and Greenability with the ISO/IEC 25010 Standard*.
- Schneier, B. (2019). *Haz clic aquí para matarlos a todos: un manual de supervivencia*. Ediciones Temas de Hoy.
- Schwaber, K., & Sutherland, J. (2020). *La Guía Scrum*.
- Scrum. (2023). *Home of Scrum!™*. <https://www.scrum.org/>
- Sobrevilla, G., Hernández, J., Velasco, P., & Soriano, S. (2017). *Aplicando Scrum y Prácticas de Ingeniería de Software para la Mejora Continua del Desarrollo de un Sistema Ciber-Físico*. 6(1), 1–15. <https://www.redalyc.org/articulo.oa?id=512253717001>

- Suaza, K. V. (2013). *Definición de equivalencias entre historias de usuario y especificaciones en UN-LENCEP para el desarrollo ágil de software.*
- Talreja, A. (2023, April 25). *Scrum Planning and Estimation: T-Shirt Sizing.* <https://teachingagile.com/scrum/psm-1/scrum-planning-estimation/estimation-techniques/t-shirt-sizing>
- Unity Technologies. (2024a). *Marco de trabajo AR Foundation de Unity | Software de realidad aumentada para desarrollo multiplataforma | Unity.* <https://unity.com/es/unity/features/arfoundation>
- Unity Technologies. (2024b). *Software para desarrollo de realidad aumentada | Motor de AR para aplicaciones | Unity.* <https://unity.com/es/unity/features/ar>
- Universidad Técnica del Norte. (2022). *UTN EN CIFRAS.* [https://cloud2.utn.edu.ec/ords/f?p=224:3::::](https://cloud2.utn.edu.ec/ords/f?p=224:3:::)
- Userlytics. (2017, November 9). *System Usability Score (SUS) and other User Testing Metrics.* <https://www.userlytics.com/resources/blog/the-system-usability-scale/>
- Vaca, T. (2017). *MODELO DE CALIDAD DE SOFTWARE APLICADO AL MÓDULO DE TALENTO HUMANO DEL SISTEMA INFORMÁTICO INTEGRADO UNIVERSITARIO – UTN.*
- Venturini, J., Garay, & Vladimir. (2021). Reconocimiento facial en América Latina. *AI Sur.*
- Wu, H. K., Lee, S. W. Y., Chang, H. Y., & Liang, J. C. (2013). Current status, opportunities and challenges of augmented reality in education. *Computers and Education*, 62, 41–49. <https://doi.org/10.1016/j.compedu.2012.10.024>