

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA INFORMÁTICA



TESIS DE GRADO

**“MODELO DE RECONOCIMIENTO FACIAL MEDIANTE
REALIDAD AUMENTADA EN APLICACIONES MÓVILES PARA
LA IDENTIFICACIÓN Y UBICACIÓN DE PERSONAS”**

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS**

POSTULANTE: LIBIA KARINA GONZALES MARIN

TUTOR METODOLÓGICO: PhD YOHONI CUENCA SARZURI

ASESOR: MSc CARLOS MULLISACA CHOQUE

LA PAZ – BOLIVIA

2016



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

Dedicado con mucho cariño a mí amado padre Ignacio por convertirse en el pilar más importante de mí vida haciendo posible este logro, a mi familia por su gran apoyo e incondicional cariño y a todos los amigos y compañeros que conocí, los cuales me acompañaron en este ciclo de la vida.

Agradecimientos

A la Universidad Mayor de San Andrés por haberme acogido en sus aulas de estudio brindando una educación de excelencia que impactó mi vida en el ámbito profesional.

A la carrera de Informática de la Facultad de Ciencias Puras y Naturales por permitirme integrar en sus filas profesionales y guiarme en el camino profesional a través de mi profesión.

Al PhD Yohoni Cuenca mi tutor metodológico y el Magister Carlos Mullisaca mi asesor, que muy profesionalmente me guiaron y apoyaron en las revisiones y correcciones de la presente tesis, por la paciencia el tiempo dedicado a el avance del presente trabajo de investigación muchas gracias.

Muchas gracias a mi familia, amigos y todas las personas que me brindaron su apoyo y conocimiento en el proceso de desarrollo de este trabajo.

Resumen

El avance diario de la tecnología y la constante investigación de esta, nos permite construir soluciones tecnológicas que se orienten a los problemas en muchos ámbitos de la sociedad, uno de estos problemas es el de la desaparición de personas en las cuales muchas veces se ve involucrada de la trata y tráfico de gente, órganos, prostitución, entre otros, al no ser encontrada ni reconocida a tiempo; es por este motivo que se trata de combinar una nueva tecnología, la realidad aumentada y utilizarla en el desarrollo de una aplicación móvil que brinde a los usuarios una forma accesible fácil y pronta de reconocer y ubicar a personas.

Se propone utilizar el reconocimiento de rostros como llave de reconocimiento, a partir de la cual se obtendrán datos registrados del mismo tanto como la ubicación y registro del mismo.

La construcción de esta aplicación fue posible con el uso de herramientas disponibles para la integración del software con la realidad aumentada y la activación de la misma a partir del reconocimiento de rostros

Los resultados obtenidos a partir de pruebas en la aplicación, el interés y aceptación de los usuarios nos indican que es posible la construcción de estas soluciones, la introducción de nuevas tecnologías y su uso al servicio de los ciudadanos.

Palabras Clave: aplicaciones móviles, realidad aumentada, reconocimiento de Rostros, búsqueda de personas.

Abstract

The constant technology research and investigation about it, allow us to build technological solutions that are oriented to society Issues, one of this is the people disappearance, it could be referenced to people traffic, human organs, prostitution among others, it if they not being found or recognized at time. this is why try to make the new technology with augmented reality that give to users an accesible, faster and agile to found, recognize and locate people.

We proposes to use the face recognition as a key to find and with it to get dates and information registred of it, as the location and registry of it

The build of this application was posible with theuse of settings that we have for the integration of software with augmented reality and activation of it with the face recognition.

The result we get making testing in the application, the aceptation and interest of usser, indicate us that is possible the build of this solutions, the introduction of the new technologies and the use of the service for citizens

Key Word: mobile apps, augmented reality, face recognition.

Contenido

CAPÍTULO 1 : INTRODUCCIÓN.....	1
1.1. Antecedentes	2
1.2. Problema	4
1.2.1. Antecedentes	4
1.2.2. Planteamiento del Problema.....	4
1.2. Objetivos	5
1.2.1. Objetivo General	5
1.2.2. Objetivos Específicos.....	5
1.3. Hipótesis.....	5
1.4. Justificación.....	6
1.4.1. Social.....	6
1.4.2. Tecnológica	6
1.4.3. Científica.....	7
1.5. Alcances y Limites	7
1.6. Metodología	8
1.6.1. Método Científico	8
CAPÍTULO 2 : MARCO TEÓRICO	11
2.1. Introducción	11
2.2. Realidad Aumentada	11
2.2.1. Fundamentos tecnológicos de la realidad aumentada	11
2.2.2. Componentes de la Realidad Aumentada.....	14
2.2.3. Ubicación GPS	15
2.2.4. Introducción a la especificación de la API de Geolocalización	18
2.2.5. SDK de Realidad Aumentada	18
2.3. Reconocimiento Facial.....	19
2.3.1. Análisis.....	20
2.3.2. Deteccion Facial.....	20
2.3.3. Herramientas para el Reconocimiento Facial.....	21
2.3.4. Descripción de Métodos Algoritmos de reconocimiento Facial	24
2.4. Tecnología Móvil	34
2.4.1. Metodologías de Desarrollo Móvil.....	34
2.4.2. Pruebas de calidad en Aplicaciones Móviles	37
2.4.3. Mobile-D.....	39
CAPÍTULO 3 : MARCO APLICATIVO	41
3.1. Introducción	41
3.2. Exploración	41
3.2.1. Establecimiento de los Usuarios.....	41
3.3. Inicialización	45
3.3.1. Establecimiento de Recursos.....	45
3.4. Producción.....	48
3.4.1. Iteración 1. Diseño de Interfaz	48
3.4.2. Iteración 3. Reconocimiento de Rostros.....	50
3.4.3. Iteración 4. Validación reconocimiento de Rostros	53
3.5. Estabilización	57
3.6. Pruebas del Sistema.....	58

3.6.1.	Pruebas en la Aplicación	58
3.6.2.	Funcionamiento de la Aplicación.....	61
CAPÍTULO 4 : ANALISIS DE DATOS Y RESULTADOS		65
4.1.	Prueba de Hipótesis	65
4.2.	Contraste de Rachas de Wald – Wolfowitz.....	65
4.3.	Toma de Decisión.....	68
CAPÍTULO 5 : CONCLUSIONES Y RECOMENDACIONES		71
5.1.	Conclusiones	71
5.2.	Recomendaciones.....	72
BIBLIOGRAFÍA		73
ANEXOS		77
Anexo A. Árbol de problemas		77
Anexo B. Árbol de objetivos		78
Anexo C. Marco Lógico		79
Anexo D: Desarrollo de la aplicación		81
Anexo F: Encuestas		87

Índice de Tablas

Tabla 2.1	Clasificación de las tecnologías de seguimiento	12
Tabla 2.2	Clase FaceRecognizer	22
Tabla 2.3	Características del desarrollo móvil.	34
Tabla 3.1	Tabla de descripción de los Usuarios involucrados	41
Tabla 3.2	Historia de usuario 1	41
Tabla 3.3	Tarea de ingeniería: Diseño de la interfaz.....	42
Tabla 3.4	Tarea de ingeniería: Integración con SDK.....	42
Tabla 3.5	Tarea de ingeniería: Reconocimiento de Rostros.....	43
Tabla 3.6	Historia de usuario 2	43
Tabla 3.7	Tarea de ingeniería: Validación Rostros Hallados y Registrados	44
Tabla 3.8	Planificación del desarrollo.....	44
Tabla 3.9	Comparación: SDK´s de realidad aumentada para la plataforma Android	46
Tabla 3.10	Planeación iteración	48
Tabla 3.11	Planeación iteración 3	50
Tabla 3.12	Caso de prueba iteración 3	52
Tabla 3.13	Planeación iteración 4	53
Tabla 3.14	Caso de prueba iteración 4	56
Tabla 3.15	Caso 1 recopilación datos para agregación de personas del tipo “Buscadas”	59
Tabla 3.16	Caso 2, recopilación de datos para agregación de personas del tipo “Buscadas”.	59
Tabla 3.17	Recuperación de datos de Persona Encontrada	60
Tabla 3.18	Pruebas referentes al tamaño rostro escaneo sin movimiento	61
Tabla 3.19	Dispositivos utilizados para las pruebas.....	61
Tabla 4.1	Esperanza y Varianza.....	66
Tabla 4.2	Propuesta de software como servicio y rango de aceptación del usuario.....	67
Tabla A.1	Tarea de ingeniería: Diseño de base de datos.....	81
Tabla A.2	Planeación iteración 2	82
Tabla A.3	Caso de prueba iteración 2	82
Tabla A.4	Planeación iteración 6	86
Tabla A.5	Caso de prueba iteración 6	86
Tabla A.6	Resultados de la encuesta ciudadana.....	89

Índice de Figuras

Figura 1.1	Diseño del sistema.	8
Figura 2.1	Gráfico de función de la tecnología GSM de ubicación.	16
Figura 2.2	Primeras siete Eigenfaces (a) y Fisherfaces (b), obtenidas con un conjunto de imágenes de la base Extended Yale Face Database B.	27
Figura 2.3	Comparación de PCA y FLD para un problema de dos clases.	28
Figura 2.4	Ejemplo del operador LBP	29
Figura 2.5	Ejemplo de los bloques utilizados para LBP	30
Figura 2.6	Código LBP de pixel central 45.....	32
Figura 2.7	Distintos tipos de vecindarios LBP.....	33
Figura 2.8	Proceso para obtener el vector descriptor LBP.....	34
Figura 2.9	Ciclo de desarrollo Mobile-D.....	39
Figura 3.1	Pantalla de inicio de la aplicación.....	49
Figura 3.2	Reconocimiento de Rostro.....	51
Figura 3.3	Registro del rostro pasos 1, 2, 3.....	51
Figura 3.4	Registro del rostro 4,5.....	52
Figura 3.5	Logo indicador de porcentaje de semejanza bajo.....	54
Figura 3.6	Logo indicador de porcentaje de semejanza medio-similar.....	54
Figura 3.7	Logo indicador de porcentaje de semejanza alto-similar.....	55
Figura 3.8	Imagen de la aplicación, discriminación Rostro “NO VALIDO” persona no registrada.....	56
Figura 3.9	Imagen de la aplicación, discriminación Rostro “VALIDO” persona registrada identificada.....	57
Figura 3.10	Diagrama de actividades: integración de la aplicación.....	58
Figura A.1	Modelo de datos.....	84
Figura A.2	Envío del mensaje de persona encontrada.....	87

CAPÍTULO 1

INTRODUCCIÓN

El tema de inseguridad ciudadana, para el caso de personas desaparecidas y el tiempo que conlleva esperar para obtener un servicio pronto efectivo de búsqueda y hallazgo, es tema de todos los días. Día con día los medios de comunicación colectiva dan a conocer hechos delictivos de secuestros, trata y tráfico además de la desaparición inexplicable de personas; Se observa en los medios de comunicación a personas informando y dando a conocer a sus propios desaparecidos, personas que desaparecieron hace días, meses e incluso años sin poder ser encontradas y sin haberse hallado alguna información eficiente de ubicación o reconocimiento para su hallazgo; Si bien algunos de estos casos pueden llegar a tener un desenlace no favorable, también pueden ser atendidas a tiempo, esto si se cuenta con un sistema que se encargue de poder realizar la llegada de información de una manera sencilla, pronta y verídica en cuanto a la información que se obtienen de las fuentes, que en algunos casos brindan información o pistas para hallar a estas personas.

En cuestión de denuncia en los caso de personas desaparecidas, se tienen ciertas normas y requisitos que implican tiempo tanto por parte del denunciante como el del proceso para poder realizarlo, en muchos casos estos podrían resultar ineficientes o no productivos, donde la principal fuente de información son las personas mismas las cuales brindan o no datos claros y acertados con los cuales se da inicio a los procesos de búsqueda y hallazgo. Esto hace que el proceso en si sea relativamente deficiente y el tiempo implicado en cada uno de los casos puede resultar una variable determinante.

Dados los avances tecnológicos que existen en la actualidad, la comunidad se ve en la posibilidad de acceder cada vez con mayor facilidad a los recursos informáticos tecnológicos inteligentes que proporcionan las nuevas tecnologías, entre estas, está la tecnología móvil que es la que mayor incidencia ha tenido en la sociedad, en Bolivia en los últimos tiempos y a la cual más del 65.1% de la población accede diariamente (CPV ,2012).

El presente documento hace el planteamiento de la problemática de seguridad ciudadana en relación al problema de personas desaparecidas o perdidas y cómo la tecnología móvil de

realidad aumentada puede ser de utilidad para coadyuvar a la problemática de reconocimiento pronto de personas y reducción de mora en los procesos de atención y denuncia temprana de este tipo de denuncias.

1.1. Antecedentes

Observatorio Nacional de Seguridad Ciudadana. El Observatorio Nacional de Seguridad Ciudadana es una Unidad Técnica dependiente del Viceministerio de Seguridad Ciudadana, con carácter interdisciplinario, concentra y recopila datos para la generación de estadísticas que orienten el análisis de la información sobre delitos, violencias y otros temas referidos a seguridad ciudadana, dirigidos a la formulación y diseño de políticas públicas en ésta materia. Esta unidad cuenta con una red social de seguridad ciudadana denominada “Ciudad Segura¹”, la cual que permite a la ciudadanía reportar hechos delictivos, violencia, accidentes de tránsito, incendios y otros de los que hayan sido víctimas o testigos, mediante la red de internet La interfaz brinda a los usuarios estadísticas, muestra todos los crímenes reportados y para denunciar crímenes cuenta con un mapa donde pueden reportar el lugar del crimen, además los usuarios pueden reportar el tipo de crimen cometido, la hora y una breve descripción del crimen. Para denunciar un hecho de crimen, ver o comentar, los usuarios simplemente deben acceder con su cuenta de Facebook o twitter que además cuenta con una aplicación para dispositivos con sistema operativo Android y Blackberry desarrollada por Dacorpsoft@2013. Esta aplicación brinda los mismos servicios de la red social en un radio de hasta 80Km de hechos criminales ya reportados (ONSC, 2013).

En la universidad nacional de Trujillo de Perú, los informáticos García y Príncipe (2014), presentaron como trabajo de graduación el desarrollo de un sistema móvil como apoyo a las comisarías en la seguridad ciudadana de la ciudad de Trujillo, la propuesta consiste en un sistema móvil, el cual permita dar soporte informático a la gestión de los servicios que brinda una comisaría y proporcionar información para la seguridad de los ciudadanos.

En una entrevista la FELCC indicó el proceso actual que se realiza en el caso de personas desaparecidas es atendida mediante la unidad de trata y tráfico, la misma consiste en el

¹ <http://www.ciudadsegura.com.bo>

registro de todos los antecedentes en una base de datos para luego realizar la impresión de un afiche el cual contiene los datos más relevantes de la persona y datos del contacto que lo busca, seguidamente se procede a la distribución de este en varios puntos de la ciudad, entre estos tranca, salidas provinciales y departamentales, se indicó además que se realizan entrevistas a familiares, amigos, vecinos entre otros para obtener información de los posibles motivos causales de su desaparición; En este proceso se detecta si el caso está relacionado con algún tipo de delito para lo cual abre el caso por “echo ilícito” y las pesquisas se proceden en ese marco (EABolivia, 2016).

Tesis y Proyectos

Aplicaciones orientadas a la seguridad ciudadana

La tesis de Alvarado (2013), titulada “Realidad aumentada en aplicaciones móviles para la seguridad ciudadana en el transporte público”, constituye el desarrollo de una aplicación móvil programada para dispositivos móviles con sistema operativo Android, la cual hace el uso de una herramienta de realidad aumentada la misma que es activada a través de códigos QR en un sistema de registro de control a Radiotaxis, donde el usuario podrá realizar un escaneo al código asignado como roseta del Radiotaxi que pretende abordar y a partir de este podrá obtener acceso a la información registrada del mismo y así poder realizar el abordaje en un medio de transporte seguro.

La tesis de Vargas (2012) titulada ” Sistema de alerta basado en tecnología móvil”, constituye el desarrollo de un sistema para dispositivos móviles con plataforma Android y para dispositivos con plataforma J2M que permite a los usuarios enviar un mensaje reportando un hecho delictivo y el tipo de hecho que esté ocurriendo en algún lugar para que estos datos se reflejen en un mapa de forma automática y se almacene en una base de datos la localización del hecho delictivo, el tipo de crimen que vio el usuario, además del IMEI del denunciante, el cual es el código pregrabado y único que tienen los teléfonos móviles, entre otros. Todo esto con el fin de lograr que la policía pueda trasladarse al lugar exacto donde haya sido reportado un hecho, además de evitar el anonimato y falsos reportes a través del IMEI de los dispositivos desde donde se envíe el mensaje.

Aplicaciones con Reconocimiento de Rostro

En la Pontificia Universidad Católica del Perú, la Ingeniera en sistemas Gonzales (2013) desarrollo como tema de Tesis de Grado en la un sistema Web y móvil los cuales permitan dar soporte informático a la gestión de los servicios que brinda una comisaría y proporcionar información para la seguridad de los ciudadanos todo esto en base a lenguaje Java y el Sistema Operativo Android.

1.2. Problema

1.2.1. Antecedentes

La ciudadanía se encuentra cada vez más insatisfecha con el lapso de tiempo que tarda una denuncia de emergencia en ser atendida, el que conlleva realizar una denuncia de desaparición de personas y el periodo que puede significar el hallazgo de las misma, esto se lo puede constatar en los diferentes medios de comunicación, las redes sociales muestran incluso linchamientos de personas, por la falta de atención y a las cuales ellos mismos “castigan” como un modo de hallar solución a una emergencia que tiene que ver con seguridad ciudadana y que no es atendida en un tiempo óptimo.

En una entrevista la FELCC informo que se registraron 180 personas de ambos sexos desaparecidas en el primer semestre del año, con edades que comprenden desde un año hasta personas de la tercera edad, esto por diversos motivos (EABolivia, 2016).

El tema de seguridad ciudadana para el caso de personas desaparecidas está ligado a distintos motivos causales y la manera actual de búsqueda y el tiempo que conlleva hallar datos de una persona puede resultar poco eficiente.

1.2.2. Planteamiento del Problema

Mejorar la seguridad ciudadana es un tema clave para Bolivia. Según la página GSMA², Solo en 2014 se registraron dos personas desaparecidas por día, 7 de cada 10 estudiantes aceptaron

² www.gsma.com

desconocidos en sus redes sociales y 10 de 10 personas han recibido alguna vez mensajes de fraude en su celular, de acuerdo a lo anterior, se plantea el siguiente problema:

“¿El reconocimiento facial mediante realidad aumentada en aplicaciones móviles reconocerá la identidad, ubicación y brindara información de las personas de manera sencilla y pronta?”.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar un modelo de reconocimiento facial aplicando realidad aumentada en dispositivos móviles para la identificación y ubicación de personas desaparecidas.

1.2.2. Objetivos Específicos

- Diseñar un modelo de reconocimiento facial mediante tecnología de realidad aumentada para dispositivos móviles con sistema operativo Android.
- Obtener y Procesar la ubicación geográfica con el manejo de SDK orientado a la realidad aumentada.
- Diseñar e implementar de una base de datos para el registro de datos agregados por la tecnología de realidad aumentada para la ubicación.
- Procesar la información recuperada en el envío de denuncia proporcionada por la tecnología de realidad aumentada.
- Modelar el Diseño e implementar pruebas sobre el prototipo de aplicación móvil.
- Aplicar el modelo en la ciudad de La Paz para que el usuario logre realizar el registro de datos de una persona y la recuperación de los mismos a través de su rostro.

1.3. Hipótesis

“El Reconocimiento facial aplicando realidad aumentada en aplicaciones móviles identifica, ubica a las personas de manera fácil y pronta en casos de personas desaparecidas”.

1.4. Justificación

1.4.1. Social

Con la inserción de tecnología móvil que involucra realidad aumentada, se busca beneficiar a la ciudadanía en cuanto a la búsqueda de personas desaparecidas, ya que este puede estar ligado a distintos problemas sociales tales como el tráfico de personas, órganos entre otros, esto para reducir el tiempo que involucra el proceso de búsqueda y hallazgo, además como un método que coadyuve a la minimización de estos problemas.

Se pretende mejorar y reducir el tiempo implicado en el proceso actual de búsqueda, hallazgo y denuncia en el campo de personas desaparecidas

“El Viceministro de Seguridad Ciudadana, Carlos Aparicio, ha pedido de instituciones y organizaciones; debido a la presunta trata y tráfico de personas, inclusive de tráfico de órganos humanos. Además, denuncian, del paso ilegal de personas que se presume que son de la república argentina, inician a operar raptando personas, de muchas de las cuales nunca más se conoce su rumbo. Asimismo, la autoridad informó que se registraron denuncias de varias mujeres desaparecidas en meses pasados, de ello no se tuvo ninguna respuesta en las investigaciones, lo que hace presumir que se trataría de un corredor de trata y tráfico de personas, incluso se sospecha tráfico de órganos humanos” (El_Diario, 2015).

1.4.2. Tecnológica

El desarrollo de aplicaciones móviles a través del tiempo va cobrando mayor importancia, en la actualidad debido al crecimiento de la demanda de dispositivos móviles Smartphone y la mayor accesibilidad a estos. Por tal razón introducir nueva tecnología con resultados en tiempo real, mayor usabilidad y accesibilidad para los usuarios puede significar la mejoría en sus requerimientos diarios a través de la tecnología. La investigación del presente texto se enfoca en la realidad aumentada, una tecnología que comienza a cobrar mayor campo en las aplicaciones móviles, cuyo concepto es sencillo y simple de entender, pero su campo de estudio es amplio con respecto al desarrollo de aplicaciones móviles.

1.4.3. Científica

La tecnología de realidad aumentada, ha ido evolucionando a través del tiempo, en la actualidad existen más de una manera de utilizar y aplicar este tipo de tecnología, en principio se tenía el código QR el cual es el modo más conocido de interfaz de realidad aumentada y uno de los primeros; El reconocimiento de imágenes es lo más reciente, el cual proporciona de un modo amigable la realidad aumentada, pues interactúa directamente con lo que es las imágenes de la vida real creando una visión futurista del mundo a través del dispositivo el cual se lo emplea; En este ámbito tenemos el reconocimiento facial, el cual aún esta perfeccionándose y pretendiendo un alcance mucho más grande en lo que a realidad aumentada se refiere.

1.5. Alcances y Limites

El sistema tendrá la limitación de tratar el reconocimiento de rostros, la recuperación de información por medio de este y la detección de la ubicación al momento del registro con las siguientes características:

- El usuario podrá registrar una persona a través de su rostro.
- Podrá otorgarle los datos de identificación, descripción y ubicación.
- Podrá registrar a una persona encontrada a través del escaneo de su rostro.
- opción de ubicación Geográfica si la activa.

El sistema tendrá los siguientes alcances:

- Reconocimiento Facial de personas desaparecidas y encontradas.
- Inserción y Obtención de datos a través de compatibilidad de rostro escaneado.
- Datos otorgados por ubicación Geográfica móvil vía GPS.
- Almacenamiento de la información en una base de datos para su uso.

La aplicación estará caracterizada como se muestra en la figura 1.1:

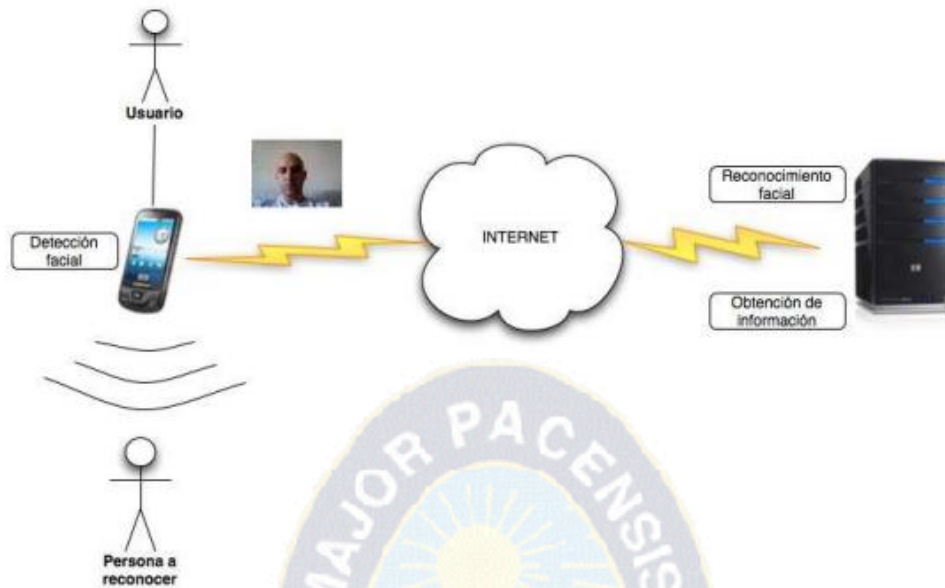


Figura 1.1 Diseño del sistema.

1.6. Metodología

1.6.1. Método Científico

El método científico es una serie ordenada de procedimientos de que hace uso la investigación científica para observar la extensión de los conocimientos obtenidos. Se puede concebir el método científico como una estructura, un armazón formado por reglas y principios coherentemente concatenados. Es quizás uno de los más útiles o adecuados, capaz de proporcionarnos respuesta a nuestras interrogantes. Respuestas que no se obtienen de inmediato de forma verdadera, pura y completa, sin antes haber pasado por el error. Esto significa que el método científico llega a nosotros como un proceso, no como un acto donde se pasa de inmediato de la ignorancia a la verdad. Este es quizás el método más útil o adecuado, ya que es el único que posee las características y la capacidad para auto corregirse y superarse, pero no el único.

Algunos caracteres principales que son parte del método científico se describen a continuación:

El Método Científico es Racional

Es racional porque se funda en la razón, es decir, en la lógica, lo cual significa que parte de conceptos, juicios y razonamientos y vuelve a ellos; por lo tanto, el método científico no puede tener su origen en las apariencias producidas por las sensaciones, por las creencias o preferencias personales. También es racional porque las ideas producidas se combinan de acuerdo a ciertas reglas lógicas, con el propósito de producir nuevas ideas.

El Método Científico es Analítico

El método científico descompone todo lo que trata con sus elementos; trata de entender la situación total en términos de sus componentes; intenta descubrir los elementos que componen cada totalidad y las interrelaciones que explican su integración. Por tal razón, los problemas de la ciencia son parciales y así con sus soluciones, más aún: los problemas son estrechos al comienzo, pero van ampliándose a medida que la investigación avanza.

El Método Científico es Claro y Preciso

La claridad y la precisión del método científico se consiguen de las siguientes formas. Los problemas se formulan de manera clara, para lo cual, se distinguen cuáles son los problemas e se incluyen en ellos los conceptos o categorías fundamentales.

El método científico inventa lenguajes artificiales utilizando símbolos y signos; a estos símbolos se les atribuye significados determinados por medio de reglas de designación.

El Método Científico Verificable

Todo conocimiento debe aprobar el examen de la experiencia, esto es, observacional y experimental. Por tal razón la ciencia fáctica es empírica en el sentido de que la comprobación de las hipótesis involucra la experiencia; pero no es necesariamente experimental y, por eso, no es agotada por las ciencias de laboratorio.

El Método Científico es Explicativo

Intenta explicar los hechos en términos de leyes, y las leyes en términos de principios; además de responder al como son las cosas, responde también a los porqués, porque suceden los hechos como suceden y no de otra manera. La explicación científica se realiza siempre en términos de leyes.

Objetivo del Método Científico

El método científico busca alcanzar la verdad fáctica mediante la adaptación de las ideas a los hechos, para lo cual utiliza la observación y la experimentación.

El método parte de los hechos intentando describirlos tales como son para llegar a formular los enunciados fácticos que se observan con ayuda de teorías se constituye en la materia prima para la elaboración teórica.

El método científico es la conquista máxima obtenida por el intelecto para descifrar y ordenar los conocimientos. Tiene los siguientes pasos a seguir:

Observación del Fenómeno, consiste en la recopilación de hechos acerca de un problema o fenómeno natural que despierta nuestra curiosidad. Las observaciones deben ser lo más claras y numerosas posible, porque han de servir como base de partida para la solución. Consiste en los siguientes puntos:

- Planteamiento del Problema.
- Formulación de la Hipótesis.
- Planteamiento de objetivos.
- Diseño del experimento.
- Obtención de resultados
- Conclusión.

2.1. Introducción

En el presente capítulo se exponen los principales conceptos y definiciones tecnológicos que representan la realidad aumentada, el reconocimiento de rostros en tecnología móvil y los métodos de desarrollo de tecnología móvil investigados para la realización de la aplicación.

2.2. Realidad Aumentada

“Para conceptualizar el término de Realidad Aumentada de manera sencilla, se hace referencia a los sentidos humanos a través de los cuales percibimos el mundo que nos rodea. Nuestra realidad física es entendida a través de la vista, el oído, el tacto y el gusto. La realidad aumentada puede describirse como un potenciador de los cinco sentidos a través de una nueva lente tecnológica por la cual la información del mundo real se complementa con la del digital” (Fundación Telefónica, 2011).

El término de realidad aumentada no es un concepto nuevo, fue acuñado por primera vez por Tom Caudell el año 1990 que describe la aumentación de la realidad física mediante el uso de técnicas que la mezclan con contenido digital virtual. La capacidad de la realidad aumentada para mezclar objetos reales y virtuales en un mismo escenario permite la utilización de objetos reales y transformarlos en virtuales o mixtos, de tal forma que es posible usar esta tecnología como una plataforma para la interacción humano-máquina (Bonnin, 2008).

2.2.1. Fundamentos tecnológicos de la realidad aumentada

La realidad aumentada como se entiende actualmente se puede considerar como una categoría de «mix reality», o de mezcla entre información real e información virtual en el que predomina la información real. El objetivo principal de esta, es el de ofrecer una experiencia de usuario en el que se diluyan las barreras entre lo real y lo virtual, y que esto se haga de una manera transparente para el usuario sin que tenga que portar complejos dispositivos. Para esto, son necesarias como mínimo tecnologías de seguimiento o «tracking» como cámaras o GPS;

tecnologías de interacción: reconocimiento de voz, reconocimiento de gestos, etc.; tecnologías de display, como pantallas transparentes o gafas. En definitiva, toda una amalgama de software y hardware de alto nivel perfectamente integrado, lo que guarda cierta similitud con las tecnologías de realidad virtual. Aunque el abanico de tecnologías necesaria para la creación de servicios inmersivos de realidad aumentada es todavía más amplia e incluye aspectos tales como Cloud Computing ³ o redes de telecomunicación ubicuas o Cloud Computing, en este apartado nos centraremos en las tecnologías más específicas de la realidad aumentada, las cuales se observan en la tabla 2.1 y que están clasificadas como tecnologías de seguimiento, de interacción y de display (Fundación Telefónica, 2011).

Tabla 2.1 Clasificación de las tecnologías de seguimiento
Fuente (Fundación Telefónica, 2011)

Tecnología	Clasificación	Descripción
Tecnologías de seguimiento	Tecnologías seguimiento basadas en sensores	Actualmente, han perdido su importancia con respecto a la utilización de cámaras de visión artificial y de los GPS. En los últimos años no ha sido nada común encontrar sistemas de «tracking» basados solamente en sensores, aunque han existido excepciones como sistemas de ultrasonidos para detectar áreas en interiores de edificios.
	Tecnologías seguimiento basada en visión	En general, las líneas de investigación relativas al seguimiento se pueden dividir en dos grandes grupos: las basadas en marcadores y las basadas en sistemas de reconocimiento de imágenes reales. De este tipo de marcadores destacan los códigos QR (Quick Response Barcode) y los códigos Bidi.
	Tecnologías seguimiento híbridas	En muchos tipos de aplicaciones, la visión por ordenador no es suficiente para realizar la actividad de seguimiento de una forma adecuada. En estos casos la visión se complementa con otras tecnologías combinadas. La tecnología de localización GPS se ha convertido en la tecnología más habitual de seguimiento que se combina con la visión, debido al auge de los Smartphone que incorporan tanto cámara como receptor

³ **Cloud Computing.** Computación en la nube, almacenamiento en la nube, nube de cómputo, informática en la nube.

		de GPS.
Tecnologías de interacción	Interfaces basadas en el uso de marcadores	En este tipo de interacción el usuario manipula un elemento real al que se ha colocado un marcador, y los resultados son reflejados en los movimientos del correspondiente objeto virtual asociado. El objeto real puede ser de muy diversas maneras, el más típico es una paleta con un marcador en la parte superior, aunque cualquier objeto es susceptible de utilizar este método. Un ejemplo de realidad aumentada es que esté basada en marcadores en el que diferentes imágenes se superponen sobre el contenido del libro cada vez que se pasa de página.
	Detección del movimiento corporal	La detección y seguimiento de una parte del cuerpo se considera la forma ideal de interacción ya que provoca en el usuario una sensación de mejor usabilidad al no tener que preocuparse por colocarse marcadores para la utilización de los servicios. Las tecnologías utilizadas en esta categoría, al igual que pasaba en los sistemas de seguimiento, se basan en el reconocimiento de imágenes por lo que tienen gran cantidad de aspectos en común.
	Interacción basada en dispositivos de bajo coste	Una alternativa muy utilizada actualmente, al menos hasta que los sistemas de visión artificial evolucionen, es la utilización de dispositivos de bajo coste para detectar el movimiento. Entre ellos podemos incluir desde el mando de la Wii, hasta sistemas médicos como el Laparoscopic Virtual Mirror, que se basa en el concepto de espejo virtual para realidad aumentada.
	Sistemas hápticos	Recientemente Investigadores de la Universidad McGill de Montreal, Canadá, han avanzado en el desarrollo de interfaces, desarrollando baldosas que pueden simular el aspecto, sonido y sensación de la nieve o del césped. Este tipo de herramientas son ideales para aplicaciones de realidad aumentada o controladores de realidad virtual. El sistema de baldosas está hecho con unas placas deformables suspendidas sobre una plataforma, y entre ambas hay sensores que detectan la fuerza ejercida por los pies del

		usuario. La placa puede emitir vibraciones que imiten la sensación de pisar diferentes materiales, e incluso añadir efecto visual y de sonido. Este suelo podría usarse incluso como una pantalla táctil controlada por los pies, como un mapa de navegación gigante en el suelo del vestíbulo de un edificio o una plaza pública. Incluso podría utilizarse para juegos añadiendo más interactividad.
Tecnologías de display		En general, el display utilizado para mostrar la información virtual fusionada con la información real es la pantalla de un ordenador o la pantalla de un teléfono móvil. De hecho, la gran revolución en las capacidades de los teléfonos móviles, más concretamente de los Smartphone que incorporan gran cantidad de tecnologías, entre ellas pantallas de una aceptable resolución, está siendo un gran revulsivo para la aparición de aplicaciones de realidad aumentada.

2.2.2. Componentes de la Realidad Aumentada

Un sistema de realidad aumentada incluye los componentes descritos a continuación (Fundación Telefónica, 2011).

- Un elemento que capture las imágenes de la realidad que están viendo los usuarios. Basta para ello una sencilla **cámara** de las que están presentes en los ordenadores o en los teléfonos móviles.
- Un elemento sobre el que proyectar la mezcla de las imágenes reales con las imágenes sintetizadas. Para ello se puede utilizar la **pantalla** de un ordenador, de un teléfono móvil o de una consola de video-juegos.
- Un **elemento de procesamiento**, o varios de ellos que trabajan conjuntamente. Su cometido es el de interpretar la información del mundo real que recibe el usuario, generar la información virtual que cada servicio concreto necesite y mezclarla de forma adecuada. Nuevamente encontramos en los PC's, móviles o consolas estos elementos.
- Un elemento al que podríamos denominar «activador de realidad aumentada» o **marcador**. En un mundo ideal el activador sería la imagen que están visualizando los usuarios, ya que a partir de ella el sistema debería reaccionar. Pero, dada la complejidad técnica que este proceso requiere, en la actualidad se utilizan otros elementos que los

sustituyen. Se trata entonces de elementos de localización como los GPS ⁴ que en la actualidad van integrados en gran parte de los Smartphone, así como las brújulas y acelerómetros que permiten identificar la posición y orientación de dichos dispositivos, así como las etiquetas o marcadores del tipo RFID⁵ o códigos bidimensionales, o en general cualquier otro elemento que sea capaz de suministrar una información equivalente a la que proporcionaría lo que ve el usuario, como por ejemplo sensores. En un caso ideal, algunos de estos elementos podrían llegar a eliminarse. Esto ocurriría si se consigue, por ejemplo, proyectar la información sintetizada de forma que el ojo sea capaz de verla, bien sobre unas gafas, directamente sobre la retina, o con alguna técnica holográfica avanzada. Pero, por el momento, esto deberíamos considerarlo todavía futurista.

2.2.3. Ubicación GPS

“La localización geográfica en Android es uno de estos servicios que, a pesar de requerir poco código para ponerlos en marcha, no son para nada intuitivos ni fáciles de llegar a comprender por completo. Todo esto no es debido al diseño de la plataforma Android en sí, sino a la propia naturaleza de este tipo de servicios, ya que por un lado, existen multitud de formas de obtener la localización de un dispositivo móvil, aunque la más conocida y popular es la localización por vía GPS, también es posible obtener la posición de un dispositivo por ejemplo a través de las antenas de telefonía móvil o mediante puntos de acceso Wi-Fi cercanos, y todos cada uno de estos mecanismos tiene una precisión, velocidad y consumo de recursos distinto. Por otro lado, el modo de funcionamiento de cada uno de estos mecanismos hace que su utilización desde nuestro código no resulte por completo directa ni intuitiva que se desearía” (Sgoliver.net, 2011).

Desde hace años, distintas instituciones en el mundo, algunas de estas relacionadas a las fuerzas de seguridad, hacen el uso de la localización de teléfonos celulares con los cuales se realizan llamadas de emergencia, los cuales envían la ubicación geográfica esto con el fin de ubicar el equipo telefónico y por ende socorrer a la persona a la que corresponde dicho dispositivo y que se encuentra en problemas.

⁴ **GPS.** Global Position System (Sistema de posición global)

⁵ **RFID.** Radio Frequency Identification (Identificación por radiofrecuencia)

No obstante, este método de localización activa es una técnica antigua para establecer la ubicación correcta de un dispositivo móvil, ya que en la actualidad se comenzó a implementar el seguimiento en modo pasivo, mediante la utilización de redes GSM.

A través del uso de las redes GSM para localizar un teléfono celular, no es necesario efectuar una llamada al teléfono móvil con el fin de poder ubicar geográficamente al dispositivo, aunque cabe destacar que el equipo debe estar encendido para que la búsqueda pueda concluir con resultados exitosos.

El método de localización por intermedio de GSM funciona de manera sencilla, ya que básicamente se trata del establecimiento de una triangulación del teléfono celular en base a su posición actual en el planeta. Se puede observar el funcionamiento de la misma en la figura 2.1.

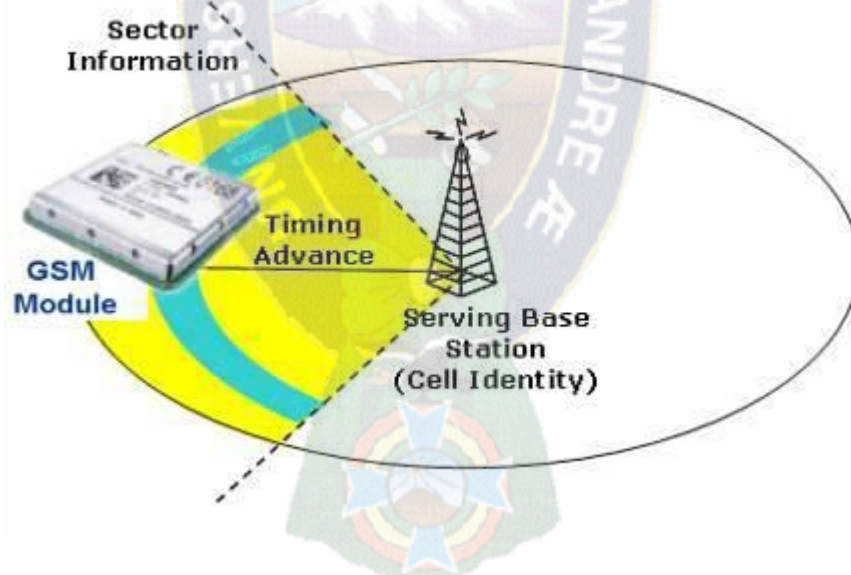


Figura 2.1 Gráfico de función de la tecnología GSM de ubicación.
Fuente (Informática-Hoy, 2015)

De la misma manera, la mayoría de las empresas del rubro operadoras de telefonía móvil, incluyen entre sus packs de servicios la prestación de localización del equipo a través de la tecnología GSM.

Para ello, el servicio se basa en distintos factores y métodos de búsqueda, esto con el fin de establecer con la mayor exactitud posible, la ubicación física de un teléfono móvil mediante la técnica pasiva.

Uno de los métodos más utilizados en la actualidad es el denominado Cell of Origin, que en español significa célula de origen, y como su nombre lo indica basan su búsqueda en la inclusión de ID de célula denominada como Cell ID e ID de célula mejorada o Enhanced Cell ID.

La técnica que utiliza la ID de célula permite obtener notables resultados de localización, que aproximadamente puede ubicar un teléfono móvil con una precisión de hasta 200 metros de radio en zonas urbanas, hasta 2 kilómetros en sectores suburbanas, y hasta de 4 kilómetros en apartados rurales.

Por su parte, el método que utiliza los datos de la ID de célula mejorada hace posible obtener mayor exactitud en la localización, sobre todo en lo que se refiere a zonas rurales, reduciendo el radio a 550 metros aproximadamente.

Existen otros métodos de localización, entre estos uno basado en la diferencia de tiempo observado, técnica denominada también como E-OTD, siglas de su nombre en inglés Enhanced-Observed Timed Difference.

En este caso, es posible lograr una precisión de entre 50 a 200 metros por cada tres estaciones base, lo cual depende directamente del número de LMUs que se encuentran disponibles en la red en ese momento.

Otros métodos utilizados muy frecuentemente suelen ser Time of Arrival, es decir tiempo de llegada, como así también Angle of Arrival o ángulo de llegada, Enhanced Observed Time Difference, es decir estimación mejorada de la diferencia de tiempo, y el sistema conocido como GPS asistido.

Este último método, que se realiza a través de la triangulación de la localización física del teléfono celular, por intermedio del uso de satélites, es una de las técnicas más implementadas en la actualidad, debido a la gran exactitud de resultados que brinda este tipo de tecnología.

Es por ello, que la mayoría de los teléfonos móviles que se comercializan en el mercado actual, sobre todo aquellos de gama media a alta, incluyen esta prestación entre sus características principales, con el fin de ofrecerle a los usuarios un método seguro para ubicar su equipo ante la posible pérdida o hurto del mismo.

2.2.4. Introducción a la especificación de la API de Geolocalización

La API de Geolocalización es un trabajo de la World Wide Web Consortium (W3C) para estandarizar una interfaz la cual logre recuperar la información de la localización geográfica de un dispositivo cliente. La API define un conjunto de objetos, que se ejecutan en la aplicación cliente para dar la ubicación del dispositivo cliente a través de la consulta de los servidores de información de la localización.

Esto se realiza por medio de una interfaz de alto nivel de información de localización asociada sólo con el dispositivo que aloja la aplicación, la información que proporciona es la siguiente: latitud, longitud, altura, rumbo, velocidad y valores de precisión de los reportes.

La API es independiente de las fuentes de información y debe ser transparente en su uso de cómo se obtuvo la localización. Algunas fuentes comunes de información sobre la ubicación incluyen los Sistema de Posicionamiento Global (GPS) y la ubicación originada de las señales de red como la dirección IP, RFID, WiFi y Bluetooth , redes GSM / CDMA de celulares, así como se puede hacer una entrada directa del usuario. Por lo anterior la API no puede garantizar devolver la ubicación real del dispositivo.

2.2.5. SDK de Realidad Aumentada

Existe una gran cantidad de SDK⁶s de realidad aumentada disponibles para la construcción de aplicaciones móviles, a continuación se describen algunos de ellos.

2.2.5.1. Metaio

Fundada en 2003 en Munich, Alemania por el actual CEO⁷ Thomas Alt y director de tecnología actual, Peter Meier. Metaio, como empresa, se crea a partir de un proyecto inicial

⁶ SDK. Software Development Kit (Kit de desarrollo de software)

con Volkswagen y posteriormente recibieron subvención alemana con la que los fundadores iniciaron su empresa. En 2005 Metaio lanzó la primera aplicación de realidad aumentada consumidor final llamada KPS Click & Design, que permitía a los usuarios colocar muebles virtuales en una imagen. Metaio luego lanzó su plataforma Unifeye que ofrecer a terceros la posibilidad de crear soluciones comerciales con realidad aumentada.

2.2.5.2. Layar.

Layar, utiliza el receptor GPS y la brújula de los teléfonos Android tales como como los HTC Dream y HTC Magic, para ubicar la posición del usuario vía GPS y su orientación o hacia dónde está mirando o dirigiéndose, con la brújula. El sensor de movimiento se utiliza para conocer el ángulo de inclinación del teléfono. La cámara del teléfono recoge el entorno y reproduce la imagen en la pantalla del teléfono, mientras que el software superpone sobre la imagen información relativa a lo que aparece en pantalla (Microsiervos, 2013).

El software dispone de diversas capas de datos con diferentes contenidos a elección del usuario. Por defecto obtiene información procedente de la capa “Layar Local Search” que utiliza datos de Google, como los utilizados en el Places Directory, para señalar la posición de lugares tales como cafeterías, restaurantes, gasolineras, centros comerciales, cines, etc. en un radio determinado entre 25 metros y 10 km aproximadamente.

La ubicación de los lugares se puede visualizar en la imagen en tiempo real del entorno o sobre un mapa. Cada posición incluye información sobre qué es, la dirección postal y la distancia a la que se encuentra. Para llegar no hay más que dirigirse hacia él manteniendo el punto en pantalla. Aunque esto último puede tener el inconveniente de la continua visualización a través de la cámara del móvil. Layar se puede descargar directa y gratuitamente desde el Android Market (layar, 2013).

2.3. Reconocimiento Facial

La identificación de características faciales ha recibido un fuerte impulso gracias al avance en la tecnología de vídeo multimedia propiciándose así un aumento de cámaras en los lugares de

⁷ CEO. Chief Executive Officer (Director ejecutivo)

trabajo, hogar y dispositivos móviles con un reducido coste. El reconocimiento facial se puede aplicar en el control de accesos a edificios públicos y privados, cajeros automáticos, laboratorios de investigación, como clave secreta de acceso para el uso de ordenadores personales o terminales móviles de última generación así como para servir de tarjeta de visita de una persona. El proceso de identificación facial se divide básicamente en dos tareas: detección y reconocimiento. La primera de ellas, la detección, comprende la localización de una o varias caras dentro de una imagen, ya sea fija o una secuencia de vídeo. La segunda tarea, el reconocimiento, consiste en la comparación de la cara detectada en el paso anterior con otras almacenadas previamente en una base de datos. Estos procesos, detección y reconocimiento, no deberían ser totalmente independientes debido a que según la forma en la que se detecte una cara puede ser prácticamente imposible su reconocimiento con caras de una base de datos detectadas de manera diferente, de ahí que los sistemas de reconocimiento facial estén fuertemente condicionados por la posición y orientación de la cara del sujeto con respecto a la cámara y las condiciones de iluminación en el momento de realizar la detección (Perez,2012).

2.3.1. Análisis

Tras realizar un análisis exhaustivo de diferentes métodos de reconocimiento y la forma de llevar a cabo dicha tarea en dispositivos con escasa capacidad de cálculo, se ha pensado que la mejor opción es hacer una división de tareas del procedimiento completo. Así, el sistema se va a dividir en 2 pilares fundamentales:

- Un dispositivo de detección
- Un servidor de reconocimiento.

2.3.2. Detección Facial

El proceso principal de detección facial se puede describir de la siguiente manera:

Existe un dispositivo de detección es el encargado de capturar la imagen o vídeo que contiene a la persona cuya cara se quiere reconocer, cuenta con una cámara, integrada o externa. Una vez capturada la imagen, detecta en ella las caras de la persona o personas a las que se esté

enfocando, se extraerá dicha región y se enviará a través de la conexión disponible en ese momento al servidor de reconocimiento base de datos (Martinez & Linares, 2012).

El proceso puede describirse con los siguientes elementos principales

- Capturar la imagen
- Detectar las caras
- Enviar la región seleccionada a un servidor de reconocimiento y esperar respuesta

Por otra parte, el servidor de reconocimiento se ejecuta en modo pasivo a la espera de recibir conexiones provenientes de los dispositivos que ejecutan el software cliente e interactúa con ellos para satisfacer las necesidades oportunas. Las tareas del servidor de reconocimiento son:

- Pre procesar la imagen que ha recibido del cliente
- Realizar el reconocimiento y obtener información extra
- Mejorar las características del clasificador
- Dar respuesta al cliente

Como puede apreciarse, las tareas del cliente son tareas livianas que podrá realizar sin consumir excesivos recursos del sistema.

2.3.3. Herramientas para el Reconocimiento Facial

2.3.3.1. OpenCV

La librería Opencv es un conjunto de clases enfocados al reconocimiento facial, está disponible para diversos sistemas operativos como Windows, Linux y Mac, no solo es multiplataforma sino que también cuenta con versiones de la librería para los lenguajes como: C/C++, Java, CSharp, Python.

2.3.3.2. Reconocimiento Facial con OpenCV

La librería Opencv cuenta con clases, una de ellas Facerecognizer, que facilitan el trabajo a la hora de hacer el reconocimiento facial, la clase facerecognizer implementa tres algoritmos:

EigenFaces, FisherFases y LBHP, de las cuales se puede implementar el que mejor se ajuste a nuestras necesidades (OpenCV.com, 2014).

Para hacer reconocimiento facial se debe realizar los siguientes pasos:

- Entrenar el algoritmo con un conjuntos de rostros capturados de la persona que deseamos reconocer.
- Obtener el rostro de la cámara, procesarlo, recortarlo y alinearlos para hacer el reconocimiento.
- Identificar a la persona aplicando un determinado algoritmo.

2.3.3.3. FaceRecognizer

Todos los modelos de reconocimiento de rostros en OpenCV son derivados de la clase base abstracta FaceRecognizer, el cual se describe en la tabla 2.2 y provee una infinidad de accesos a todos los algoritmos en OpenCV.

Tabla 2.2 Clase FaceRecognizer
Fuente (OpenCV, 2014)

```
class FaceRecognizer : public Algorithm
{
public:
    /// virtual destructor
    virtual ~FaceRecognizer() {}

    /// Trains a FaceRecognizer.
    virtual void train(InputArray src, InputArray labels) = 0;

    /// Updates a FaceRecognizer.
    virtual void update(InputArrayOfArrays src, InputArray labels);

    /// Gets a prediction from a FaceRecognizer.
    virtual int predict(InputArray src) const = 0;

    /// Predicts the label and confidence for a given sample.
    virtual void predict(InputArray src, int &label, double &confidence) const = 0;

    /// Serializes this object to a given filename.
    virtual void save(const string& filename) const;
```



```

// Deserializes this object from a given filename.
virtual void load(const string& filename);

// Serializes this object to a given cv::FileStorage.
virtual void save(FileStorage& fs) const = 0;

// Deserializes this object from a given cv::FileStorage.
virtual void load(const FileStorage& fs) = 0;

// Sets additional information as pairs label - info.
void setLabelsInfo(const std::map<int, string>& labelsInfo);

// Gets string information by label
string getLabelInfo(const int &label);

// Gets labels by string
vector<int> getLabelsByString(const string& str);
};

```

En principio FaceRecognizer no se ve como una interfaz de gran alcance. Sin embargo: Cada FaceRecognizer es un algoritmo, por lo que puede conseguir/cambiar fácilmente, todos los elementos internos modelo esto si es permitido por la aplicación. Algoritmo es un concepto relativamente nuevo OpenCV, que está disponible desde la versión 2.4

Algoritmo proporciona las siguientes características para todas las clases derivadas:

- El llamado "constructor virtual". Es decir, cada derivado del algoritmo se ha registrado al inicio del programa y se puede obtener la lista de algoritmos registrados y crear instancia de un algoritmo determinado por su nombre como se observa en el algoritmo. Si se añaden algoritmos propios, resulta práctico el añadir prefijos a los creados para distinguirlos de otros algoritmos.
- Configuración / Recuperación de parámetros del algoritmo por su nombre. Si ha utilizado la funcionalidad de captura de vídeo desde el módulo highgui OpenCV, está probablemente familiarizado con cvSetCaptureProperty (), cvGetCaptureProperty (), VideoCapture :: set () y VideoCapture :: get (). Algoritmo proporciona un método similar donde en lugar de enteros de id especifica los nombres de los parámetros como

cadena de texto. Los métodos `:: set ()` y el Algoritmo `:: get ()` se observan en la tabla 2.2 para más detalles.

- Lectura y escritura de parámetros desde ficheros XML o YAML. Cada algoritmo derivado puede almacenar todos sus parámetros y los volverá a leer. No hay necesidad de volver a aplicar cada vez.

Además cada FaceRecognizer apoya la:

- La formación de un FaceRecognizer con `FaceRecognizer :: tren ()` en un determinado conjunto de imágenes, su base de datos la cara.
- Predicción una imagen de muestra dada de, que significa una cara . La imagen se administra como una estera.
- Cargar / guardar el estado del modelo de / a un XML o YAML dado.
- Ajuste / Obtención de información de etiquetas, que se almacenan como una cadena, esta Cadena de etiquetas de información es útil para mantener los nombres de las personas reconocidas (OpenCV, 2015).

2.3.4. Descripción de Métodos Algoritmos de reconocimiento Facial

Descripción de los métodos El proceso de reconocimiento de caras consiste en tomar una imagen de dos dimensiones, a filas y b columnas, a la que se transforma en un vector unitario contenido en un espacio de imágenes n-dimensional $n = a \times b$. Luego se le sustrae la imagen promedio y se proyecta el vector resultante en un sub espacio de menor dimensión utilizando uno de los métodos de reducción de dimensión, la extracción de características. Esta proyección es comparada con la proyección de un conjunto de imágenes de una base. La clase del vector más similar, utilizando algún criterio de similitud, es el resultado del proceso de reconocimiento.

Existen diversos algoritmos que a través del tiempo fueron desarrollados con bases matemáticas con el objetivo de optimizar el reconocimiento de rostros en diferentes aplicaciones (Ottado, 2014).

Estos algoritmos se describen a continuación:

2.3.4.1. Algoritmo Eigenfaces

Este método realiza una proyección lineal del espacio de imágenes a un espacio de características de menor dimensión. Esta reducción se realiza utilizando la técnica PCA la cual toma aquella proyección lineal que maximiza la dispersión de todas las imágenes proyectadas (Ottado, 2014).

En primer lugar se considera a un conjunto de N imágenes con valores en el espacio de imágenes n -dimensional

$$x_i \quad i = 1, 2, \dots, N \quad (1)$$

Se asume además que cada una de las imágenes pertenece a una de las c clases $\{x_1, x_2, \dots, x_c\}$. Asimismo se considera una transformación lineal que lleva el espacio de imágenes original de n dimensiones al espacio de características de dimensión m , donde $m < n$. Los nuevos vectores de características $y_k \in \mathfrak{R}^m$ son definidos por la siguiente transformación lineal

$$y_k = W^T x_k \quad k = 1, 2, \dots, N \quad (2)$$

Donde $W \in \mathfrak{R}^{n \times m}$ es una matriz con columnas ortonormales. Se define además la matriz de distribución total S_T como:

$$S_T = \sum_{k=1}^N (x_k - \mu)(x_k - \mu)^T \quad (3)$$

Donde $\mu \in \mathfrak{R}^n$ es la media de todas las imágenes de (1). Luego de aplicar la transformación lineal W^T , la distribución de los vectores de características $\{y_1, y_2, \dots, y_N\}$ es $W^T S_T W$. Se toma aquella proyección W_{opt} que maximiza el determinante de la distribución total de la matriz de las imágenes proyectadas, esto es:

$$W_{opt} = \text{arg} \max_w W^T S_T W = [w_1 w_2 \dots w_m] \quad (4)$$

Donde $\{w_i \mid i = 1, 2, \dots, m\}$ es el conjunto de vectores propios n -dimensionales de S_T correspondiente a los mayores m vectores propios. Dichos vectores propios tienen la misma

dimensión que las imágenes originales y se les denomina eigenfaces. En la Figura 2.2 se muestra las primeras siete eigenfaces obtenidas con un conjunto de imágenes de la base Extended Yale Face Database B.

2.3.4.2. Algoritmo Fisherfaces

El método Fisherfaces utiliza el Discriminante Lineal de Fisher mas conocido como FLD para la reducción de dimensión. Este método selecciona el W de la ecuación (1) de manera que el cociente entre la distribución entre clases y la distribución intra-clases sea máxima (Ottado, 2014).

Para esto se define la matriz S_B de distribución entre clases como

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (5)$$

Y la matriz S_W de distribución intra-clases

$$S_W = \sum_{i=1}^c \sum_{x_k \in X_i} N_i (x_k - \mu_i)(x_k - \mu_i)^T \quad (6)$$

Donde μ_i es la imagen media de la clase X_i , y N_i es el número de imágenes en la clase X_i . Si la matriz S_W es no singular, la proyección W_{opt} se elige como la matriz con columnas ortonormales que maximiza el cociente del determinante de la matriz de distribución entre clases de las imágenes proyectadas y el determinante de la matriz de la distribución intraclases de las imágenes proyectadas, esto es:

$$W_{opt} = \underset{W}{\operatorname{arg\,max}} \frac{W^T S_B W}{W^T S_W W} = [w_1, w_2 \dots w_m] \quad (7)$$

Donde $\{w_i \mid i = 1, 2, \dots, m\}$ es el conjunto de valores propios de S_B y S_W correspondiente a los m mayores valores propios $\{\lambda_i \mid i = 1, 2, \dots, m\}$, esto es:

$$S_B w_i = \lambda_i S_W w_i \quad i = 1, 2, \dots, m \quad (8)$$

Se observa entonces, que a lo sumo se tienen $c - 1$ valores propios distintos de cero, y por lo tanto el límite superior de m es $c - 1$, donde c es el número de clases. Para el problema de

reconocimiento de caras, se tiene que la matriz $SW \in \mathfrak{R} \text{ n} \times \text{n}$ es siempre singular, dado que el rango de SW es a lo sumo $N - c$, y en general, el número de imágenes de entrenamiento: N , es mucho más chico que el número de pixeles de cada imagen: n . Por lo tanto puede ser posible elegir una matriz W tal que la distribución intra-clases de las imágenes proyectadas pueda ser exactamente cero. Como alternativa entonces, al criterio establecido en la ecuación (7), se proyecta el conjunto de imágenes a un espacio de menor dimensión, de manera que la matriz resultante de la distribución intra-clases SW es no singular. Utilizando PCA se realiza la reducción de dimensiones del espacio de características a $N - c$ y luego, aplicar FLD definido en (7) para reducir la dimensión a $c-1$. De esta manera W_{opt} es dado por:

$$W_{pca}^T = W_{fld}^T W_{pca}^T \quad (9)$$

Dónde:

$$w_{pca} = \arg_w^{max} W^T S_T W$$

$$w_{fld} = \arg_w^{max} \frac{W^T W_{pca}^T S_B W_{pca} W}{W^T W_{pca}^T S_w W_{pca} W}$$

Las columnas de esta matriz se les refieren como fisherfaces. La Figura 2.2 muestra las primeras siete fisherfaces obtenidas con un conjunto de imágenes de la base Extended Yale Face Database B.

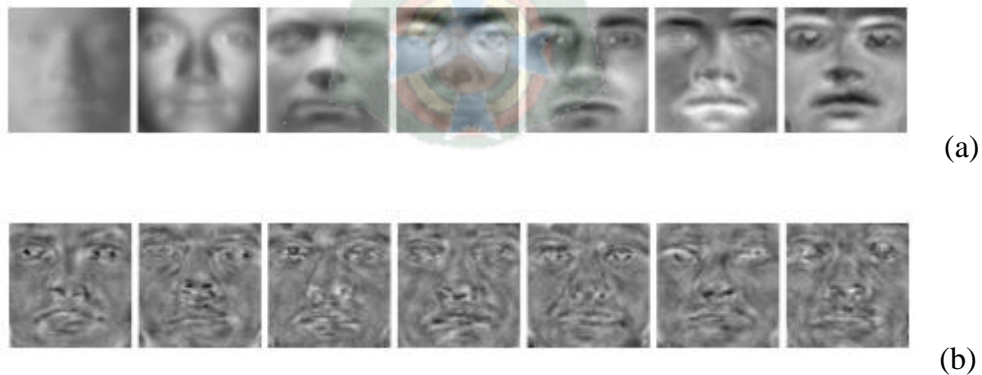


Figura 2.2 Primeras siete Eigenfaces (a) y Fisherfaces (b), obtenidas con un conjunto de imágenes de la base Extended Yale Face Database B.

Fuente (Ottado, 2014)

En la Figura 2.3 se ilustra la ventaja de FLD sobre PCA para el caso de un problema de clasificación en el que las muestras de cada clase yacen próximas a un sub espacio lineal. El problema consiste de dos clases en el cual las muestras de cada clase han sido ligeramente desplazadas en una dirección perpendicular al sub espacio lineal, por lo que cada clase yace próxima a una línea que pasa por el origen en el espacio de características de dos dimensiones. Ambos métodos, PCA y FLD, han sido utilizados para proyectar las muestras en un espacio de una dimensión. Se observa que PCA proyecta las muestras de manera que las clases no puedan ser linealmente separables. Por lo tanto FLD consigue una mayor dispersión entre clases.

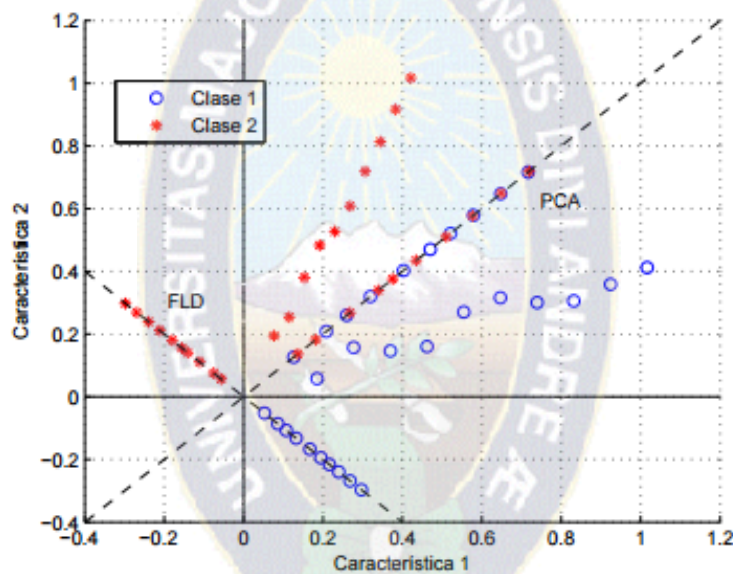


Figura 2.3 Comparación de PCA y FLD para un problema de dos clases.
Fuente (Ottado, 2014)

2.3.4.3. Algoritmo Local Binary Pattern (LBP)

El operador original LBP es una herramienta interesante como descriptor de textura. Este operador etiqueta los píxeles de una imagen umbralizando una vecindad de 3x3 respecto al valor del píxel central y considerando el resultado como un número binario. Luego el histograma de las etiquetas puede utilizarse como descriptor de textura. En la figura 2.4 se ilustra el funcionamiento del operador LBP básico (Aguerreberre & Capdehourat, 2015).

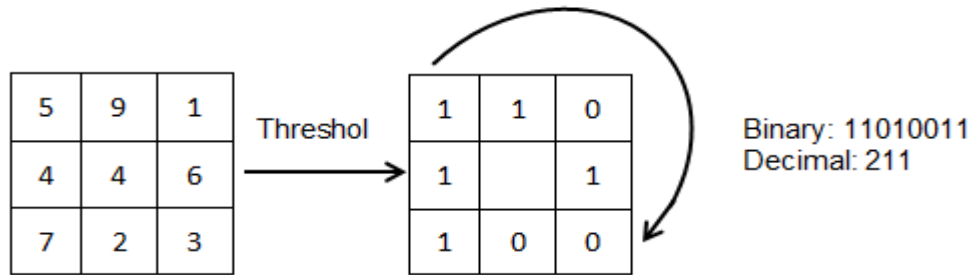


Figura 2.4 Ejemplo del operador LBP
Fuente (Aguerrebere & Capdehourat, 2015)

Algunas variantes al operador básico fueron introducidas más adelante. Una de ellas es el hecho de trabajar con entornos de tamaño variable. Usando vecindades circulares e interpolando los valores de los píxeles, cualquier radio es válido.

Otra extensión es el hecho de tomar en cuenta solamente los patrones uniformes, siendo _estos los que tienen al menos 2 transiciones 0-1 o 1-0. Esto se apoya en el hecho de que la mayor parte de los patrones en una imagen son patrones uniformes. En la Figura 2.4 se puede ver más en detalle este operador.

Aplicación a caras

El histograma obtenido mediante el procedimiento descrito contiene información acerca de la distribución local de los micro-patrones, como bordes, puntos y otros, sobre la imagen completa. Para tener una representación eficiente de la cara es necesario además almacenar información espacial según Aguerrebere et al. (2015).

En la figura 2.4 se divide en regiones uniformes mediante una grilla cuadrangular en bloques 7x7. Luego se calculan los histogramas para cada bloque y se concatenan todos para obtener el patrón de la cara.

De esta forma se logra efectivamente una descripción de la cara en 3 niveles diferentes de ubicación: las etiquetas contienen información de los patrones a nivel de los píxeles, los histogramas en una pequeña región producen información a nivel regional y los histogramas regionales son concatenados para crear una descripción global de la cara.

Para poder utilizar este descriptor aprovechando los puntos marcados, se realizó una modificación a lo hecho en la figura 2.4 En vez de usar todos los bloques de la imagen, se optó por usar solamente cuatro, pero definidos por los puntos obtenidos para los ojos, la nariz y la boca. De esta manera los histogramas se calculan siempre en zonas relevantes de la cara.

Se puede ver un ejemplo de los bloques que se obtienen en la figura 2.5 Se debe aclarar que estos bloques son de tamaño variable porque dependen de los puntos marcados manualmente, pero los histogramas obtenidos son de tamaño fijo.

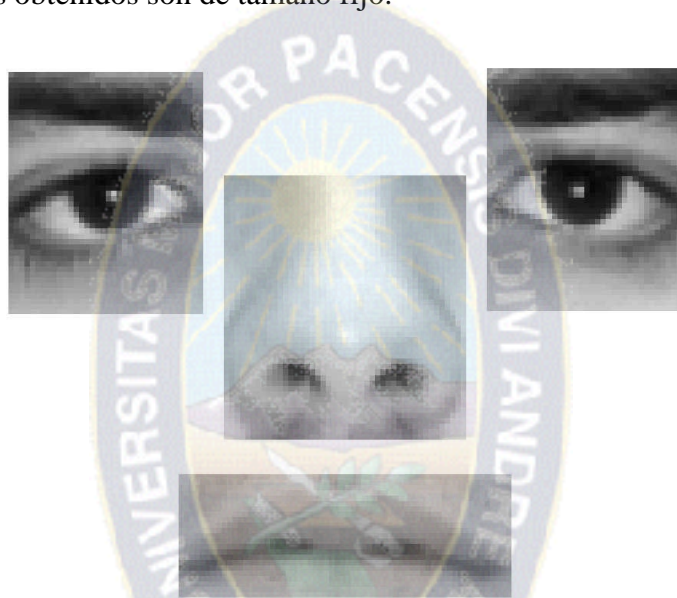


Figura 2.5 Ejemplo de los bloques utilizados para LBP
Fuente (Aguerrebere & Capdehourat, 2015)

Como tamaño de las vecindades se trabaja con 3 radios diferentes: 1, 2 y 3. El largo de cada histograma es de 10, 18 y 26 respectivamente. Concatenando los 4 histogramas se obtiene el patrón correspondiente a la cara.

Histogramas:

$$H_k = \{h_1, \dots, h_{k+2}\} \quad \text{con } k=8,16 \text{ y } 24$$

Patrones:

$$H^{total} = \{H^{ojo_i}, H^{ojo_d}, H^{nariz}, H^{boca}\}$$

$$H_k^{tot} = \{H_1^{oi}, \dots, H_{k+2}^{oi}, H_1^{od}, \dots, H_{k+2}^{od}, H_1^n, \dots, H_{k+2}^n, H_1^b, \dots, H_{k+2}^b\}$$

$$H_k^{tot} = h_1, \dots, h_{4x(k+2)}$$

Clasificación de patrones de LBP

Para el caso de los patrones obtenidos mediante LBP la medida de similitud utilizada es una de las que se definen en la figura 2.4 basada en la estadística X^2 .

Para dos vectores A y B la medida es:

$$X^2_{A,B} = \frac{(A_i - B_i)^2}{A_i + B_i}$$

Si se considera que cada región genera un vector y que los patrones son la concatenación de los vectores generados, la distancia entre patrones queda:

$$X^2_{A,B} = \frac{(A_{i,j} - B_{i,j})^2}{A_{i,j} + B_{i,j}}$$

Dónde: j es el índice en las regiones

i es el índice en cada vector.

Esto permite dar un peso distinto a cada región según la relevancia que se le quiera dar. La expresión de la distancia con pesos queda:

$$X^2_{w_j A,B} = \sum_{i,j} w_j \frac{(A_{i,j} - B_{i,j})^2}{A_{i,j} + B_{i,j}}$$

Operador de textura LBP

El operador de textura LBP, trabaja con un vecindario dentro de un marco de dimensiones 3x3, en el que se toman los valores de los ocho píxeles vecinos a este y en el cual se sitúa como el umbral al valor del pixel central de este marco para la asignación binaria. A

continuación los valores binarios se ponderan por potencias de dos y se suman para obtener el código LBP del píxel central. La figura 2.6 muestra un ejemplo del operador LBP (Zúñiga, 2015).



Figura 2.6 Código LBP de píxel central 45
Fuente (Zúñiga, 2015)

Si se definen g_c y g_0, \dots, g_7 que denotan respectivamente a los valores en escala de grises del centro y de los píxeles del entorno, el código de LBP para el píxel central con coordenadas “x” e “y” se calcula:

$$LBP_{x,y} = \sum_{p=0}^7 s(g_c - g_p) * 2^p$$

Donde s es la función de umbral definida como:

$$S(z) = \begin{cases} 1 & \text{si } z \geq 0 \\ 0 & \text{si } z < 0 \end{cases} \quad (1)$$

En el ejemplo de la figura 2.6 se tendría:

$$LBP(45) = 1 * 2^0 + 0 * 2^1 + 0 * 2^2 + 0 * 2^3 + 0 * 2^4 + 1 * 2^5 + 1 * 2^6 + 0 * 2^7 \quad (2)$$

$$LBP(45) = 1 + 8 + 3 + 64 = 105 \quad (3)$$

La extensión es capaz de tomar cualquier radio y vecinos alrededor de un píxel central, denotado por $LBP_{p,r}$, mediante el uso de un vecindario circular y la interpolación bilineal cada vez que el punto de muestreo no cae en el centro de un píxel.

Por ejemplo, $LBP_{16,2}$ hace referencia a 16 vecinos en un vecindario de radio 2.

En la figura 2.7 se muestran ejemplos con diferentes vecinos y radios.

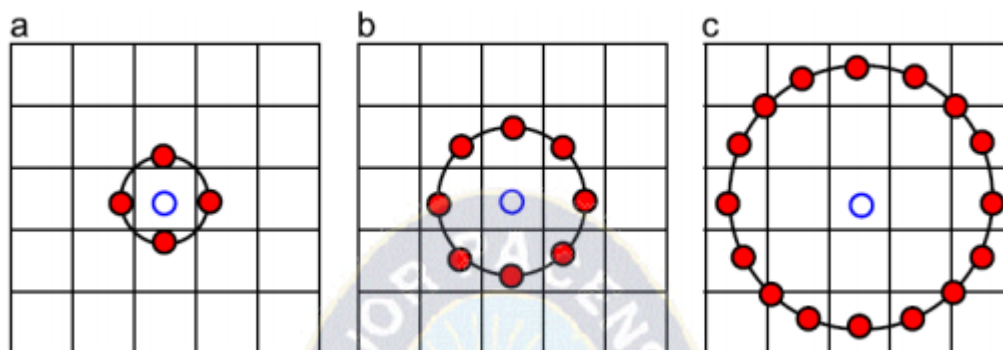


Figura 2.7 Distintos tipos de vecindarios LBP
Fuente (Zúñiga, 2015)

En la figura 2.7 se describen los siguientes puntos:

- Se muestra un $LBP_{4,0}$ de 4 vecinos y radio 0, en la figura 2.7.
- Se muestra un $LBP_{8,1}$ de 8 vecinos y radio 1 y en la figura 2.7.
- Se muestra un $LBP_{16,2}$ de 16 vecinos y radio 2.

Conociendo la aplicación del operador LBP en un punto de la imagen, es necesario conocer el desarrollo del almacenamiento y ordenación de la información en este operador para poder comprender el proceso de uso del mismo.

En el caso del método LBP el elemento que se obtiene al aplicarlo a una imagen es un vector. Dicho vector es el resultado de la concatenación de los histogramas que se obtienen al tomar la imagen y sub-dividir esta en ventanas. A cada punto en la ventana se le aplica el operador LBP dándole así un valor a cada pixel. A continuación de este proceso se generan los histogramas para cada una de las ventanas y finalmente estos histogramas proceden a concatenarse para dar origen al vector descriptor resultante de la imagen.

En la figura 2.8 se puede apreciar el proceso descrito, en el cual se observan 3 elementos del vector resultante y la imagen principal dividida en ventanas.

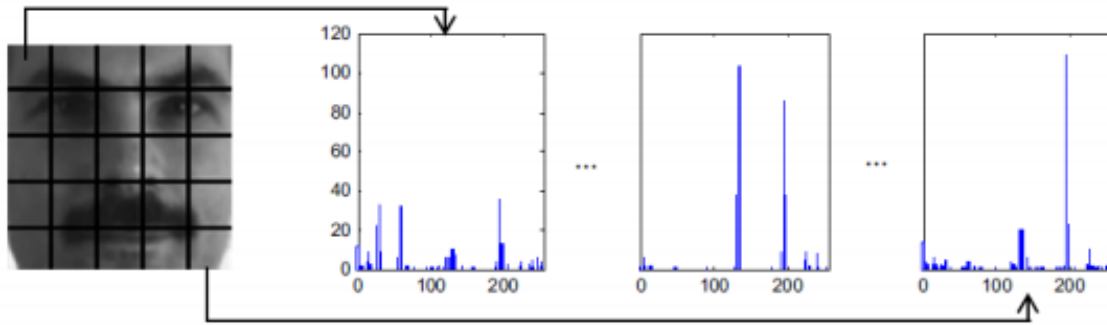


Figura 2.8 Proceso para obtener el vector descriptor LBP

Fuente (Zúñiga, 2015)

2.4. Tecnología Móvil

La tecnología móvil consiste en la utilización de medios informáticos, sin la necesidad de estar emplazados, es decir que pueden ser utilizados desde cualquier parte.

Una aplicación móvil es un programa que ha sido desarrollado para dispositivos móviles cuya función es la de realizar alguna tarea específica. Su instalación y acceso se da específicamente a través de dispositivos móviles (Viterbo, 2014).

2.4.1. Metodologías de Desarrollo Móvil

Según el artículo escrito por Blanco et al. (2009), el desarrollo de aplicaciones móviles difiere del desarrollo de software tradicional en muchos aspectos, esta diferencia provoca que las metodologías usadas para estos entornos también difieran de las del software clásico. Esto es porque el software móvil tiene que satisfacer una serie de requerimientos y condicionantes especiales que lo hace más complejo, las condiciones y requerimientos para este tipo de desarrollo se ven reflejadas en la tabla 2.3 con las respectivas descripciones y características.

Tabla 2.3 Características del desarrollo móvil.

Fuente (Blanco, Camero, Warterski & Rodriguez, 2009)

Condición/Requerimiento	Característica
Canal radio	Consideraciones tales como la disponibilidad, las desconexiones, la variabilidad en el ancho de banda, la heterogeneidad de redes o también los riesgos de seguridad han de tenerse especialmente

	en cuenta en este entorno de comunicaciones del tipo móvil.
Movilidad.	Aquí influyen consideraciones como la migración de direcciones, alta latencia debido a cambio de estación base o la gestión de la información dependiente de localización. Sobre esta última, de hecho, se pueden implementar un sinnúmero de aplicaciones, pero la información de contexto asociada resulta muchas veces incompleta y varía frecuentemente.
Portabilidad	La característica portabilidad de los dispositivos terminales implica una serie de limitaciones físicas directamente relacionadas con el factor de forma de los mismos, como el tamaño de las pantallas, o del teclado, limitando también el número de teclas y su disposición y para el caso de dispositivos sin teclado es la capacidad multi-touch del dispositivo.
Fragmentación de la industria	La existencia de una considerable variedad de estándares, protocolos y tecnologías de red diferentes añade complejidad al escenario del desarrollo móvil.
Diseño	Desde el punto de vista del desarrollo, el diseño multitarea y la interrupción de tareas es clave para el éxito de las aplicaciones de escritorio; pero la oportunidad y frecuencia de éstas es mucho mayor que en el software tradicional, debido al entorno móvil que manejan, complicándose todavía más debido a la limitación de estos dispositivos
Usabilidad.	Las necesidades específicas de amplios y variados grupos de usuarios, combinados con la diversidad de plataformas tecnológicas y dispositivos, hacen que el diseño para todos se convierta en un requisito que genera una complejidad creciente difícil de acotar donde la interfaz con el usuario debería ser sencilla y accesible no siendo así en todos los casos.
Time-to-market.	En un sector con un dinamismo propio, dentro de una industria en pleno cambio, los requisitos que se imponen en términos de tiempo de lanzamiento son muy estrictos y añaden no poca dificultad en la gestión de los procesos de desarrollo.

2.4.1.1. Metodologías Ágiles

A continuación, se muestra lo que podrían ser las características básicas de una metodología ideal para el proceso de desarrollo de software para plataformas móviles (Blanco, Camero, Werterski & Rodriguez, 2009).

Agilidad. Las metodologías ágiles mejoran la flexibilidad del desarrollo y la productividad, proveyendo métodos que se adaptan a los cambios y que aprenden de la experiencia.

Conciencia de mercado. El mercado actual está orientado hacia los productos software por lo que un proceso de desarrollo móvil debería enfocarse al desarrollo del producto y no del proyecto. Consecuentemente, los procesos deben enfocarse al nicho del negocio haciendo un análisis del mercado y toda esta información mitigará las dudas y los riesgos. Procesos orientados al mercado seguirán una planificación bastante estricta a lo que se refiere a requerimientos de time-to-market (Rahimian & Ramsin, 2008):

Soporte a toda la línea de producción software. Se refiere al "conjunto de sistemas intensivos de software compartiendo un conjunto de características comunes que satisfacen las necesidades de un segmento particular del mercado y que son desarrolladas con una serie de valores centrales en una forma predeterminada" (Clemments & Northrop, 2003). Esto hace que el ciclo de vida sea más corto, pudiendo desarrollar una familia de productos software móviles en un solo intento, reduciendo costes. Debe tener especial importancia la línea de producción para la mejora de la calidad del software.

Desarrollo basado en arquitectura. La eficiencia de la línea de producción de software depende del desarrollo de una plataforma común, por lo que la necesidad de una arquitectura genérica para una clase de productos es esencial, pudiendo reconfigurarse de forma específica para cada componente o producto determinado.

Soporte para reusabilidad. El desarrollo basado en componentes y el basado en capas ahorran costes de desarrollo, agiliza la entrega del producto y hace el software menos propenso a errores ya que los componentes no deben ser hechos desde cero cada vez.

Inclusión de sesiones de revisión y de aprendizaje. La metodología debería incorporar sesiones de revisión en todo el proceso para asegurar el análisis del producto y sesiones de aprendizaje después de la entrega de cada producto para que la experiencia sea analizada y registrada, y así la abstracción del conocimiento obtenido realimente a todo el equipo.

2.4.2. Pruebas de calidad en Aplicaciones Móviles

Desde el inicio de su construcción o desarrollo es necesario probar las aplicaciones móviles teniendo en cuenta, precisamente, su propio contexto de movilidad y considerando el cumplimiento de las características de funcionalidad, usabilidad, seguridad y rendimiento de las mismas. La complejidad del aseguramiento de la calidad del software en aplicaciones móviles radica en la existencia de diferentes sistemas operativos y multitud de modelos de dispositivos. De esta forma, una aplicación debidamente construida y de calidad será aquella que funcione correctamente en todos los dispositivos móviles y bajo todos los sistemas operativos existentes en el mercado.

“Las pruebas funcionales deberían llevarse a cabo considerando las características propias de estos dispositivos. La aplicación debe ser capaz de guardar, tanto de forma efectiva como selectiva, la información, dejando en el servidor aquellos datos que precisen de recursos adicionales. Finalmente, es necesario observar el comportamiento y fallos de la aplicación cuando pueda verse afectada por una situación de batería baja, entrada de llamadas o mensajes mientras realizamos una operación con la misma“ (Cuera, 2013).

Algunas pruebas funcionales o de caja negra a considerar son (QActions, 2013):

- Test de integración
- Test del sistema
- Pruebas de regresión
- Test de usabilidad
- Test de aceptación de usuarios

“Hablando de la usabilidad de una aplicación, es necesario poner especial atención en el desarrollo de pruebas que permitan asegurar que el diseño e interfaz resultan sencillos e

intuitivos, así como que el flujo de comunicación entre la App y el usuario se establece de forma natural y lógica” (Cuera, 2013).

En cuanto a las pruebas técnicas se realizan las siguientes (QActions, 2013):

Pruebas de Carga o stress. Se simula el acceso en simultáneo de múltiples usuarios a un servidor y medimos el comportamiento del sistema a medida que se llegan a los umbrales esperados, hasta estresar el sistema. Este tipo de pruebas deben ser llevadas a cabo con herramientas de Load Test.

“Verse afectada por una situación de batería baja, entrada de llamadas o mensajes mientras realizamos una operación con la misma” (Cuera, 2013).

Algunas pruebas funcionales a considerar son (QActions, 2013):

- Test de integración
- Test del sistema
- Pruebas de regresión
- Test de usabilidad
- Test de aceptación de usuarios

“Hablando de la usabilidad de una aplicación, es necesario poner especial atención en el desarrollo de pruebas que permitan asegurar que el diseño e interfaz resultan sencillos e intuitivos, así como que el flujo de comunicación entre la App y el usuario se establece de forma natural y lógica” (Cuera, 2013).

Pruebas de Compatibilidad. Se realizan estas pruebas para asegurar que el software sea compatible con distintas versiones de sistemas operativos, web servers, base de datos y otros componentes que constituyen la arquitectura de la aplicación.

Finalmente, señalar que la conexión a Internet es también una consideración a tener en cuenta en lo que a pruebas de software se refiere. Esta conexión permite realizar búsquedas de información desde cada dispositivo, en cualquier momento y desde cualquier lugar. En este sentido, es necesario prestar atención a los caracteres especiales de escritura de los diferentes

idiomas, las monedas de cada país, los formatos de códigos postales, fechas, direcciones, números de teléfono, entre otros, que deberían visualizarse dependiendo de la zona geográfica en la que se encontrara el usuario.

Es necesario dedicar especial atención a las pruebas de software desde las primeras fases de su desarrollo, ya que esto ayudará a las empresas a posicionar sus aplicaciones con una valoración de cinco estrellas, que otorga la máxima calidad en los market/stores de aplicaciones móviles. Y en el caso de las webs optimizadas y aplicaciones web, se garantizará su calidad, evitando el abandono de los usuarios.

2.4.3. Mobile-D

El objetivo de este método es conseguir ciclos de desarrollo muy rápidos en equipos muy pequeños. Fue creado en un proyecto finlandés en 2005, pero sigue estando vigente. Basado en metodologías conocidas pero aplicadas de forma estricta como: Extreme Programming, Crystal Methodologies y Rational Unified Process (Rodríguez, 2011).

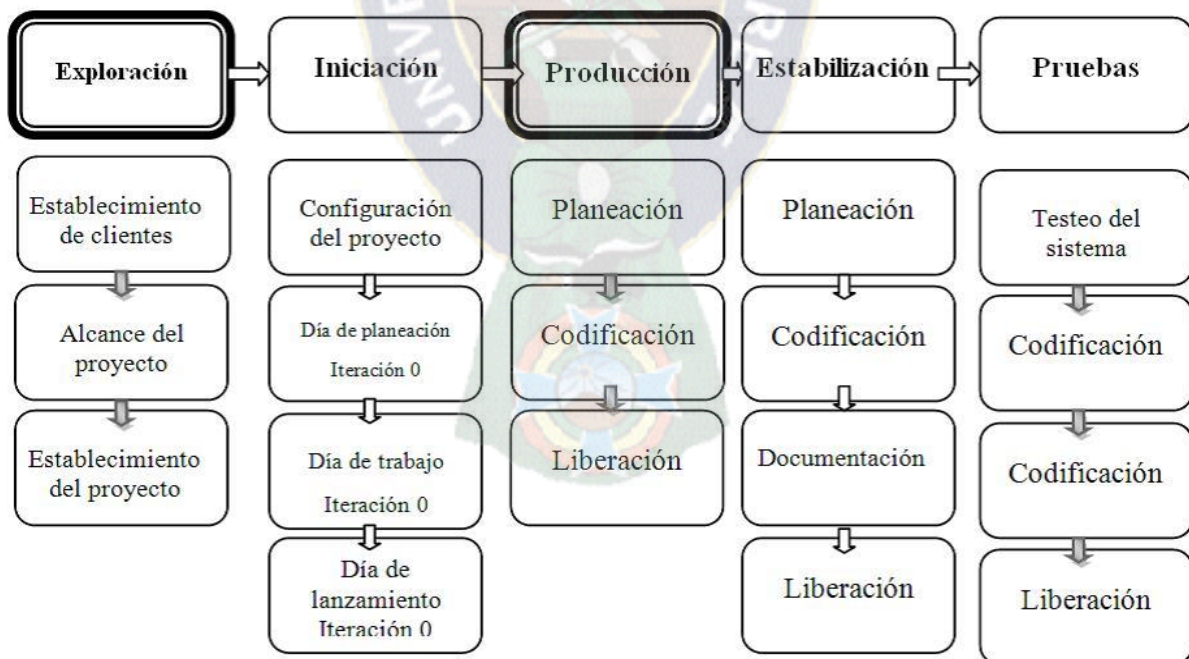


Figura 2.9 Ciclo de desarrollo Mobile-D
Fuente (González, Remis & Fernández, 2011)

Se compone de distintas fases indicadas en la figura 2.9 los cuales son: exploración, inicialización, fase de producto, fase de estabilización y la fase de pruebas. Cada una tiene un día de planificación y otro de entrega.

En la fase de exploración se centra la atención en la planificación y a los conceptos básicos del proyecto. Aquí es donde se hace una definición del alcance del proyecto y su establecimiento con las funcionalidades donde se quiere llegar.

En la iniciación se configura el proyecto identificando y preparando todos los recursos necesarios como se señaló anteriormente a esta fase se dedica un día a la planificación y el resto al trabajo y publicación.

En la fase de producción se repiten iterativamente las sub fases. Se usa el desarrollo dirigido por pruebas TDD por sus siglas en inglés, antes de iniciar el desarrollo de una funcionalidad debe existir una prueba que verifique su funcionamiento. Se puede decir que en esta fase que se lleva a acabo toda la implementación.

Después de la fase de producto llega la fase de estabilización en la que se realizan las acciones de integración para enganchar los posibles módulos separados en una única aplicación.

Fase de pruebas. Una vez parado totalmente el desarrollo se pasa una fase de testeo hasta llegar a una versión estable según lo establecido en las primeras fases por el cliente. Si es necesario se reparan los errores, pero no se desarrolla nada nuevo.

Una vez acabada todas las fases se debe tener una aplicación publicable y entregable al cliente.

3.1. Introducción

El desarrollo de creación de la aplicación se enmarcó en la metodología Mobile-D, una metodología orientada exclusivamente al desarrollo de aplicaciones móviles, en este capítulo se describe el desarrollo del prototipo en cada una de sus fases.

3.2. Exploración

3.2.1. Establecimiento de los Usuarios

Para este proceso se definirán los usuarios involucrados en el sistema los cuales se describen en la tabla 3.1.

Tabla 3.1 Descripción de los Usuarios involucrados

Cientes/Usuarios	Descripción
Usuario de la Aplicación Móvil	Persona que hará el uso de la aplicación en su dispositivo móvil, la cual podrá registrar personas encontradas y desaparecidas
Administrador de Información	Persona entidad administradora de la información del registro y administración de personas desaparecidas y encontradas.

3.2.1.1. Historias de Usuario y Tareas

En este proceso se describen las diferentes tareas que se podrán realizar en el sistema, la historia del usuario 1 es descrito en la tabla 3.2 con sus respectivas características de aplicación.

Tabla 3.2 Historia de usuario 1

Historia de Usuario Registrar	
Número: 1	Nombre Historia de Usuario Registrar: Registrar personas desaparecidas y encontradas, a través de la cámara del

	dispositivo.
Modificación de Historia de Usuario Número : -	
Usuario: Usuario móvil	Iteración Asignada: 1-3
Prioridad en Sistema: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Media	Puntos Reales: 3
Descripción: Visualizar imágenes físicas o fotografías de personas que sean consideradas "desaparecidas" o "Encontradas" mediante de la aplicación en tiempo real	
Observaciones:	

El diseño de la interfaz como tarea de ingeniería es descrito en la tabla 3.3 con sus respectivas características de aplicación.

Tabla 3.3 Tarea de ingeniería: Diseño de la interfaz

Tarea de Ingeniería	
Número Tarea: 1.1	Número Historia de Usuario Registrar: 1
Nombre Tarea: Diseño de la interfaz	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra	Puntos Estimados: 1
Fecha Inicio: 05-03-2016	Fecha Fin: 07-03-2016
Programador Responsable: Libia Gonzales	
Descripción: Diseño de la interfaz de inicio de la aplicación	

La integración con SDK de realidad aumentada es descrita en la tabla 3.4 con sus respectivas características de aplicación.

Tabla 3.4 Tarea de ingeniería: Integración con SDK

Tarea de Ingeniería Integración con SDK	
Número Tarea: 1.2	Número Historia de Usuario: 1
Nombre Tarea: Integración con SDK de realidad aumentada	
Tipo de Tarea : Desarrollo - Integración Desarrollo / Corrección / Mejora / Otra	Puntos Estimados: 2
Fecha Inicio: 08-03-16	Fecha Fin: 18-03-2016

Programador Responsable: Libia Gonzales
Descripción: Utilizar el SDK elegido para activar la realidad aumentada en la aplicación

La tarea de ingeniería para el reconocimiento de Rostros es descrita en la tabla 3.5 con sus respectivas características de aplicación.

Tabla 3.5 Tarea de ingeniería: Reconocimiento de Rostros

Tarea de Ingeniería	
Número Tarea: 1.3	Número Historia de Usuario: 1
Nombre Tarea: Reconocimiento de reconocimiento de Rostros	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra	Puntos Estimados: 3
Fecha Inicio: 19-03-16	Fecha Fin: 21-03-2016
Programador Responsable: Libia Gonzales	
Descripción: Reconocer y procesar en tiempo real los rostros de personas.	

La historia del usuario 2 es descrita en la tabla 3.6 con sus respectivas características de aplicación.

Tabla 3.6 Historia de usuario 2

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Validación de Personas Encontradas a través de la cámara móvil de la aplicación
Modificación de Historia de Usuario Número: -	
Usuario: Usuario	Iteración Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimados:
Riesgo en Desarrollo: Alto	Puntos Reales:
Descripción: Verificar rostros y datos obtenidos al realizar el escaneo de personas para ser enviados al administrador mediante la aplicación para su verificación.	
Observaciones: ninguna	

La tarea de ingeniería de validación de rostros hallados y registrados es descrita en la tabla 3.7 con sus respectivas características de aplicación.

Tabla 3.7 Tarea de ingeniería: Validación Rostros Hallados y Registrados

Tarea de Ingeniería	
Número Tarea: 2.1	Número Historia de Usuario: 2
Nombre Tarea: Validación Rostros Hallados y Registrados	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra	Puntos Estimados:
Fecha Inicio: 22-03-2016	Fecha Fin: 28-03-2016
Programador Responsable: Libia Gonzales	
Descripción: Mostrar un icono que revele el grado porcentaje de reconocimiento para personas En registro, y de envío para registro de Hallazgo	

3.2.1.2. Planificación del Desarrollo

En la tabla 3.8 se describe la planificación realizada para cada una de las historias de usuario en el punto anterior

Tabla 3.8 Planificación del desarrollo.

ITERACIONES	Nº	Tareas	INICIO	FIN
Primera	1	Diseño de la interfaz	05-03-16	07-03-16
	2	Integración SDK r.a.	8-03-16	18-03-16
	3	Reconocimiento de Rostros	19-03-16	21-03-16
Segunda	1	Validación Rostros	22-03-16	28-3-16
Tercera	1	Diseño e implementación de base de datos	02-04-16	03-04-16
	2	Realización del registro de inserción y registro de hallazgo	06-04-16	15-04-16
	3	Investigación a partir del registro de rostro reconocido valido	16-04-16	25-04-16

3.3. Inicialización

En esta fase se realizó la preparación y definición de los recursos y herramientas utilizadas en la construcción del software.

3.3.1. Establecimiento de Recursos

En este punto se describirán las herramientas utilizadas para la implementación de la aplicación.

3.3.1.1. Plataforma Android

La elección de esta plataforma fue influenciada en gran medida por el gran uso actual en el mercado, además de su confiabilidad con usuarios desarrolladores, especialmente en la ciudad de La Paz, además se procedió a las pruebas y la consulta de funcionalidad para los recursos de la PC utilizada en el proceso de desarrollo, por lo que se utilizaron las siguientes herramientas:

- Eclipse ADT (Android Development Tools)
- Android SDK Tools v.22.0.1
- Android SDK Platform Tools v.17

3.3.1.2. Elección del SDK de realidad aumentada.

Con el propósito de la construcción del prototipo, en la presente investigación se tuvo la oportunidad de realizar pruebas con distintos SDK's de realidad aumentada brindados por diferentes empresas y laboratorios y centros de investigación que están enfocados a esta tecnología.

Después de la descarga y prueba de distintos SDK's se observaron las ventajas y desventajas en el proyecto para la elección final del SDK, teniendo en cuenta además las limitaciones de máquina ya que la aplicación fue desarrollada en una PC con procesador core i3, estas ventajas y desventajas halladas en la comparación es descrita a continuación en la tabla 3.9 acorde a cada SDK utilizado.

Tabla 3.9 Comparación: SDK´s de realidad aumentada para la plataforma Android

SDK	VENTAJAS	DESVENTAJAS
<p>VUFORIA (http://developer.qualcomm.com/dev/augmented-reality)</p>	<p>Soporte para plataformas Android y iOS, reconocimiento de marcadores, proyección de objetos 3D, licencia libre y comercial. Buen soporte de la comunidad, tutoriales y documentación. Excelente opción para renderización de imágenes con el uso de OpenGL mediante marcadores.</p>	<p>Divide el código entre el lenguaje Java y lenguaje nativo, por lo tanto, para la plataforma Android se necesita descargar el NDK para la compilación de los códigos nativos. Esto significa que para el uso de sus librerías y sus ejemplos se deben seguir una serie de pasos antes de poder crear un proyecto Android y utilizar el SDK. Para crear un proyecto deben incluirse ambos lenguajes, creando archivos adicionales y añadiendo líneas de código a las clases y el uso del NDK de Android es obligatorio.</p>
<p>METAIO (http://www.metaio.com)</p>	<p>Muy popular para campañas de publicidad de distintas empresas (Wolkswagen , Adidas, etc.). Soporte para plataformas Android, iOS, web, Mac y Windows. Servicios en la nube, servidores propios. Soporte GPS, IMU, marcadores, imágenes, códigos QR, proyecciones 3D, licencia libre y comercial (incluye licencia perpetua y para los desarrolladores el uso ilimitado de la versión para PC del creador de Metaio). Utiliza AREL, (<i>Augmented Reality Experience Language</i>), una unión de JavaScript API del SDK Metaio en combinación con una definición de contenido estático XML y HTML5. Excelente soporte por la comunidad, página actualizada constantemente para desarrolladores, noticias laboratorios de investigación, concursos para desarrolladores e investigadores.</p>	<p>Para la plataforma Android, se requiere la versión mínima Froyo 2.2 y el uso del SDK está desactivado en los dispositivos que utilizan 6 chipset ARM como el HTC Hero, Magia, Wildfire y Tadoo, Samsung Galaxy Ace y dispositivos similares. Para el desarrollo de la aplicación se cuenta con un dispositivo Samsung S3 mini, en el cual las pruebas son imposibles.</p>

<p>JUNAIO (http://www.junaio.com/develop/)</p>	<p>Junaio es la versión que Metaio lanzó al mercado como “<i>The Most Advanced AR Browser</i>”, el buscador de realidad aumentada. Incluye todos los beneficios de Metaio, incluyendo las del desarrollador</p>	<p>Las desventajas son las mismas que las descritas para Metaio, además los ejemplos deben ser descargados y probados a través del browser, lo que implica mayor procesamiento del dispositivo.</p>
<p>DROIDAR (http://code.google.com/p/droidar/) (https://github.com/bitstars/droidar)</p>	<p>Proyecto de realidad aumentada todavía bajo desarrollo, sin embargo las versiones lanzadas incluyen realidad aumentada basada en localización y marcadores. Es posible contribuir a este proyecto que actualmente se encuentra en Github, y que tiene gran soporte de la comunidad debido a que es un proyecto de código libre. Tiene muchas características entre las que se incluye el renderizado de imágenes y proyecciones 3D. La realidad aumentada por geo localización incluye el uso de mapas e información en tiempo real.</p>	<p>La desventaja de este proyecto de realidad aumentada, está en el hecho que todavía es un proyecto bajo construcción. Su uso, bajo marcadores no incluye el procesamiento de códigos QR, y el renderizado es todavía un poco tosco en la presentación y proyección de imágenes ya que utiliza mayormente clases nativas del lenguaje Java y un poco de OpenGL.</p>
<p>LAYAR</p>	<p>Soporte para plataformas Android y iOS. Muy solicitado para campañas publicitarias, permite el desarrollo de estas campañas en su plataforma, dando al desarrollador gran soporte y revisión del trabajo. Soporta uso de objetos 3D, objetos multimedia, imágenes, audio, video, y realidad aumentada basada en geo localización. A partir de la versión 7.2 da soporte al procesamiento de códigos QR. Su licencia es tanto comercial como de prueba de 30 días. Cuenta con gran documentación de las distintas versiones del SDK, fórum y blog para desarrolladores. Los desarrolladores de layar brindan soporte inmediato ante un pedido o pregunta.</p>	<p>Solamente la realidad aumentada basada en marcadores permite el uso de geo localización. No es posible aumentar capas de información encima de las que son brindadas por el SDK.</p>

El SDK elegido fue Laya SDK v.7.2., puesto que las características expuestas en la tabla 3.9 incluyendo el escaneo y procesamiento de elementos de realidad aumentada y el servicio de ubicación GPS. Así también para el reconocimiento de Rostros se hizo el uso de la librería OpenCV, el cual utiliza los algoritmos de reconocimiento Fisherfaces, Eigenfaces y LBP, descritos en el capítulo anterior.

3.4. Producción

En esta fase se describe las iteraciones para la construcción de la aplicación siguiendo la regla de la metodología MOBILE-D: “programación de tres días”. A continuación se describen 3 iteraciones principales, las demás iteraciones pueden ser referenciadas en el Anexo 5.

3.4.1. Iteración 1. Diseño de Interfaz

[Historia de usuario 1, tarea de ingeniería 1.1]. La primera iteración describe el diseño de la interfaz de inicio de la aplicación.

3.4.1.1. Planeación.

La fase de planeación de la iteración se describe a continuación en tabla 3.10.

Tabla 3.10 Planeación iteración

CARACTERISTICAS	DURACIÓN
En esta fase se diseñó y desarrolló la interfaz de inicio de la aplicación en el cual se muestra al usuario un botón de linterna, un botón de activación de Ubicación, la pantalla de la cámara, un botón para el uso de entrenamiento de registros, el botón de búsqueda y el botón de cambio de cámara.	3 días

3.4.1.2. Codificación

La interfaz de inicio es una actividad en la programación Android, en el que se pueden observar los diferentes botones para registro “Entrenar” y un botón que inicia reconocimiento de los rostros en modo búsqueda “Buscar”.

Además se tiene en la parte superior de la pantalla:

- Un botón de encendido de linterna.
- Activación de la ubicación geográfica, el cual se recuperan los datos de latitud, longitud y precisión geográfica(esta opción funciona si la opción de GPS se encuentra activada).
- La pantalla de la cámara la cual estará en un escaneo constante a partir del proceso que se le indique según la opción principal elegida.
- La imagen del icono de la aplicación.

En la parte inferior de la pantalla:

- Botón que realizara el intercambio de cámara entre el frontal y el normal.

3.4.1.3. Liberación

En la figura 3.1 se muestra la pantalla de inicio en la etapa de liberación.

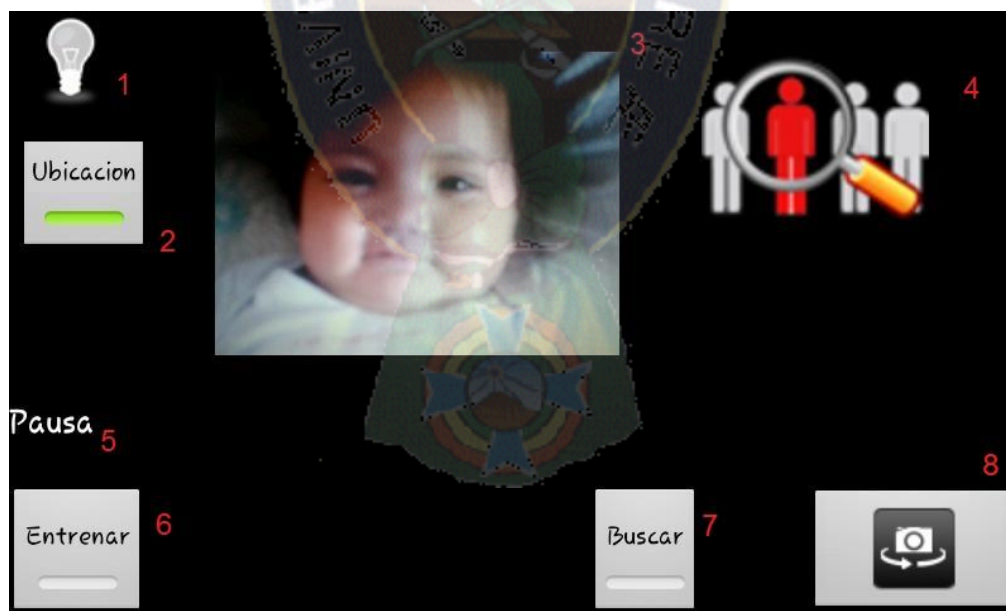


Figura 3.1 Pantalla de inicio de la aplicación

Se describen los siguientes elementos en la figura 3.1

1. Botón linterna.

2. Botón de activación de Ubicación.
3. La pantalla de la cámara para el escaneo.
4. Logo de la aplicación.
5. Estado de la cámara.
6. Botón de inicio de Registro “Entrenar”.
7. Botón de inicio de búsqueda en escaneo “Buscar”.
8. Cambio de cámara frontal principal

3.4.2. Iteración 3. Reconocimiento de Rostros

[Historia de usuario 1, tarea de ingeniería 1.3]. La tercera iteración describe el reconocimiento de Rostros en la aplicación.

3.4.2.1. Planeación

La fase de planeación de la iteración se describe a continuación en tabla 3.11.

Tabla 3.11 Planeación iteración 3

CARACTERISTICAS	DURACIÓN
La aplicación toma los Rostros como marcadores activadores de la realidad aumentada.	3 días

3.4.2.2. Codificación

La codificación en esta iteración cuenta con dos nuevas actividades Android incluidas en el SDK de layar, mediante las cuales se realizan el procesamiento de ubicación y reconocimiento de los Rostros con apoyo de la librería Open CV. La actividad que reconoce y captura el rostro contiene un widget de tipo ImageView, en el cuál se captura el rostro mostrándolo al momento del escaneo. La segunda actividad se muestra mediante dos widget, uno de tipo EditText, que recopilara los datos que se tienen para enviarlos en conjunto al rostro escaneado y otro del tipo TextView que indicara el tipo de dato debe introducirse en los EditText correspondientes a los mismos.

La figura 3.2 muestra el reconocimiento de rostros realizado por el posicionamiento de la cámara en el rostro, en el cual se puede observar que los rostros identificados son enmarcados con un cuadro de bordes verde alrededor.

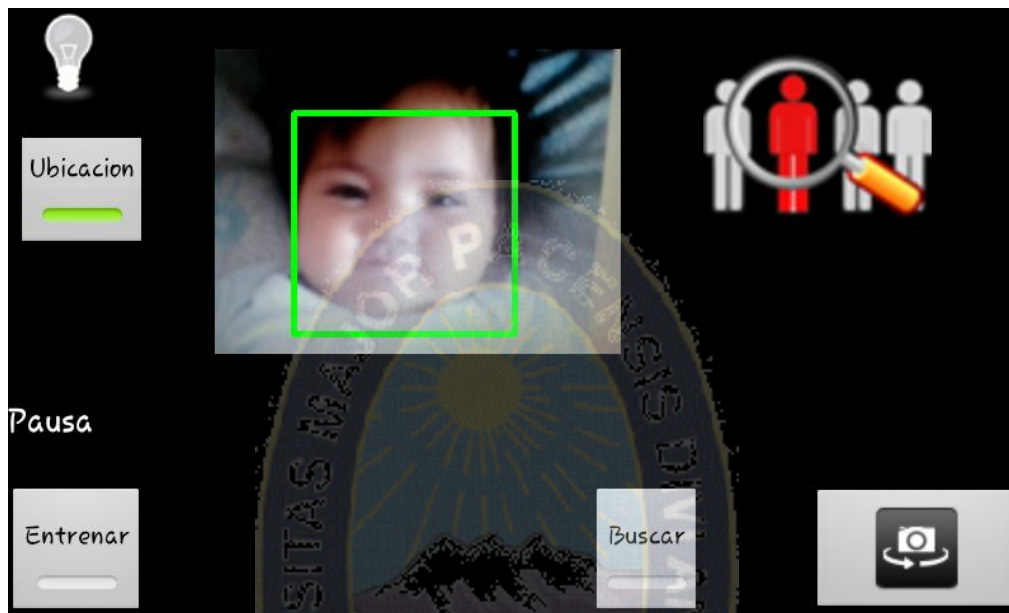


Figura 3.2 Reconocimiento de Rostro

En la figura 3.3 se muestra el registro del rostro realizado al hacer la pulsación en el botón de registro “Entrenar”.

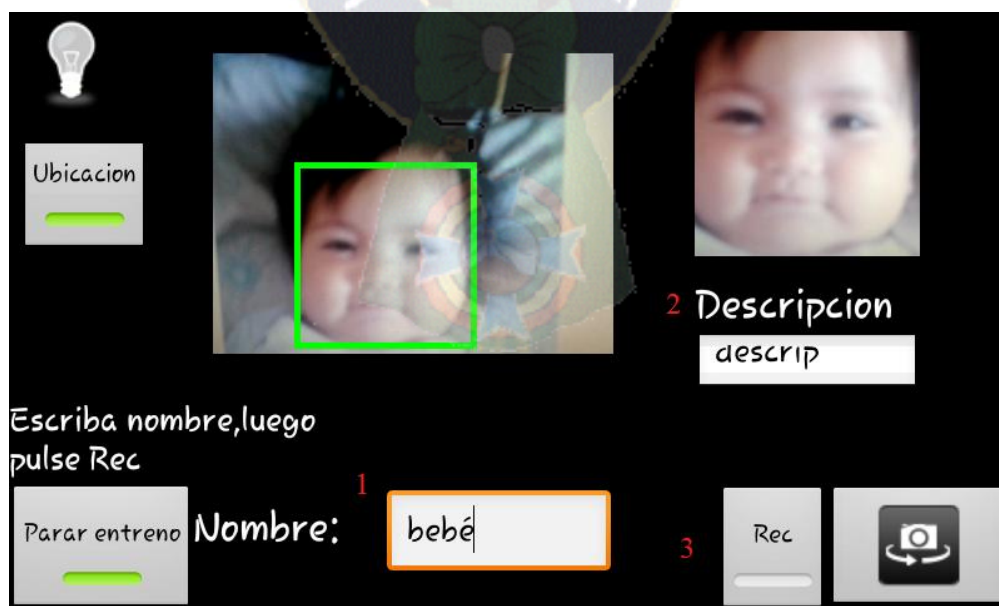


Figura 3.3 Registro del rostro pasos 1, 2, 3

A continuación se describen los pasos del proceso de registro indicados en la figura 3.3.

1. Se registra el nombre de la persona.
2. Se registra la descripción (puede ser datos relevantes y/o números de referencia)
3. Se graba el rostro (Este es mostrado en la parte donde se encontraba del icono de la aplicación)

En la figura 3.4 se muestra el proceso para concluir el envío de los datos y el rostro registrado.

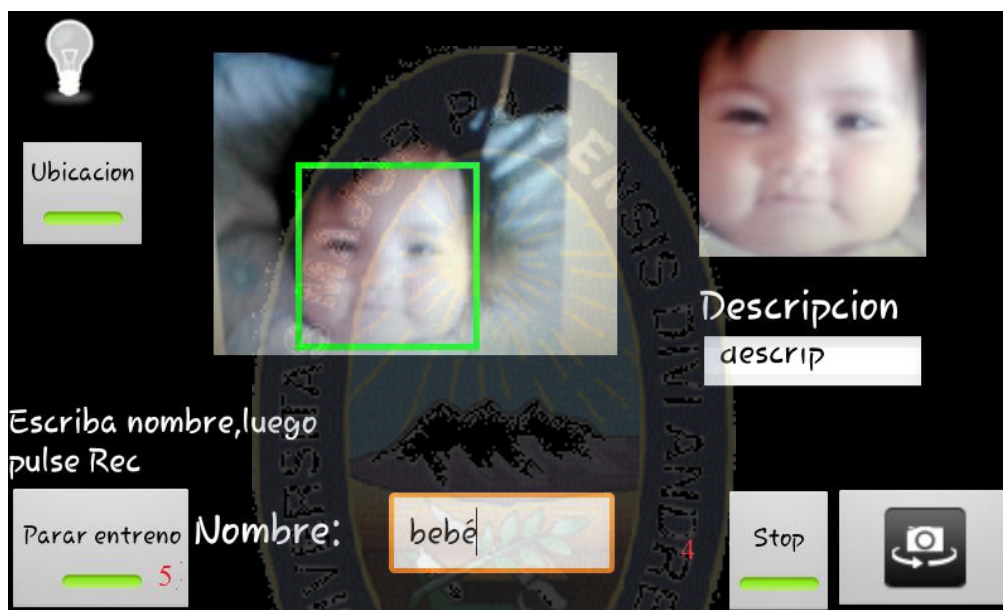


Figura 3.4 Registro del rostro 4,5

Se indican las siguientes dos funciones finales de registro:

4. Fin de la Grabación
5. Se para el entreno de Búsqueda y al mismo tiempo se concluye el registro.

3.4.2.3. Liberación

A continuación en la tabla 3.12 se describe las pruebas realizadas sobre esta iteración.

Tabla 3.12 Caso de prueba iteración 3

Caso de Prueba de Aceptación	
Código Caso de Prueba: 2	Número Historia de Usuario: 1
Descripción de la Prueba:	
Prueba de la aplicación con el reconocimiento Rostros y ubicación	

Condiciones de Ejecución: La aplicación debe reconocer los Rostros, seguidamente se tendrá la opción de agregación de ubicación.
Entrada / Pasos de ejecución: Rostro de fotografías y físicas, redes wi-fi, etc.
Resultado Esperado: Identificación del Rostro con el mensaje e icono de reconocimiento.
Evaluación de la Prueba: La aplicación realizó exitosamente la prueba como se muestra en la figura 3.4.

3.4.3. Iteración 4. Validación reconocimiento de Rostros

[Historia de usuario 2, tarea de ingeniería 2.1]. La cuarta iteración describe el proceso de validación del reconocimiento de Rostros, en el cual una persona es escaneada para obtener los datos de la misma si es que hubiese sido registrado.

3.4.3.1. Planeación

La fase de planeación de la iteración se describe a continuación en tabla 3.13.

Tabla 3.13 Planeación iteración 4

CARACTERISTICAS	DURACIÓN
Se filtraron los Rostros por el atributo “Tipo” en la base de datos para validar y diferenciar los que son buscados y hallados para que puedan diferenciar, estos datos serán devueltos al realizarse el escaneo para buscar a una persona y al ingresar a un rostro a la base de datos.	3 días
Al momento de realizar el proceso de “búsqueda” se expondrán logos que indican el porcentaje de la validez referente a la semejanza en el momento del reconocimiento mediante escaneo.	3 días

Para el grado de aceptación y reconocimiento del rostro se tienen iconos de tipo semáforo en colores Rojo, Amarillo y Verde los cuales se muestra en la parte inferior del layout de la cámara del dispositivo móvil.

Los iconos se muestran y describen a continuación:

Icono Semáforo Rojo

La visualización del icono mostrado en de la figura 3.5 nos indica que el rostro y la base principal de rostros no coinciden, por lo que la persona escaneada, puede no encontrarse registrada como buscada.



Figura 3.5 Logo indicador de porcentaje de semejanza bajo

Icono Semáforo Amarillo

La visualización del icono mostrado en la figura 3.6 indica que el rostro escaneado y la base principal de Rostros tiene un porcentaje de parecido del 80%, por lo cual podría encontrarse en la base principal, misma que al ser escaneada mostrara los datos de la persona a la cual asemeja parecido.



Figura 3.6 Logo indicador de porcentaje de semejanza medio-similar

Icono Semáforo Verde

La visualización del icono mostrado en la figura 3.7 indica que el rostro escaneado y la base principal de rostros tiene un porcentaje de parecido del 90%-100%, por lo cual se encuentra en la base principal, misma que al ser reconocida mostrara los datos de la persona a la cual asemeja el rostro.



Figura 3.7 Logo indicador de porcentaje de semejanza alto-similar

3.4.3.2. Codificación

La codificación de esta característica toma la iteración 3 como base de reconocimiento y procesamiento de rostros, se realiza la validación de los Rostros tomando como válidos aquellos rostros aceptados por el administrador de la base de datos que previamente fueron registrados, los cuales pueden ser “Buscado” o “Encontrado”, se muestra al usuario un icono de reconocimiento de tipo semáforo en color verde o amarillo y se indican los datos correspondientes a la persona escaneada

Para el caso de que la persona no sea reconocida por la App o no se encuentre dentro de la base de datos, la aplicación mostrara un mensaje indicando que la persona escaneada es desconocida, por lo cual podría no estar registrada en el sistema o no cumple con las características para un reconocimiento óptimo. Ambos casos son pueden ser vistos en el punto 3.4.3.3. Liberación.

3.4.3.3. Liberación

La fase de liberación de la iteración se describe a continuación en tabla 3.14, en la cual se realiza la prueba para el caso de la validación en el reconocimiento del rostro

Tabla 3.14 Caso de prueba iteración 4

Caso de Prueba de Aceptación	
Código Caso de Prueba: 3	Número Historia de Usuario: 2
Descripción de la Prueba: La aplicación de discriminar Rostros escaneados en modo búsqueda, entre los que serán identificados como “VALIDOS”, y aquellos que serán los “NO VALIDOS”	
Condiciones de Ejecución: Deben existir distintos tipos de rostros reconocidos incluyendo aquellos válidos e inválidos	
Entrada / Pasos de ejecución: Escaneo del Rostro o también fotografías.	
Resultado Esperado: Mostrar en pantalla el icono y el mensaje respectivo a la validez o no del código.	
Evaluación de la Prueba: La prueba exitosa es reflejada con la figura 3.9 y la prueba fallida reflejada con la figura 3.8.	

En la figura 3.8 se muestra el proceso de reconocimiento para un rostro que no se encuentra registrado en la base de datos describiremos este como “No Valido”, esto a partir de la inicialización del proceso de búsqueda activado con el botón “Buscar”. En este proceso se muestran los siguientes datos recuperados:

1. Nombre desconocido (Unknow)
2. Descripción desconocida (Unknow)
3. Icono de reconocimiento semáforo color rojo (NoValido)

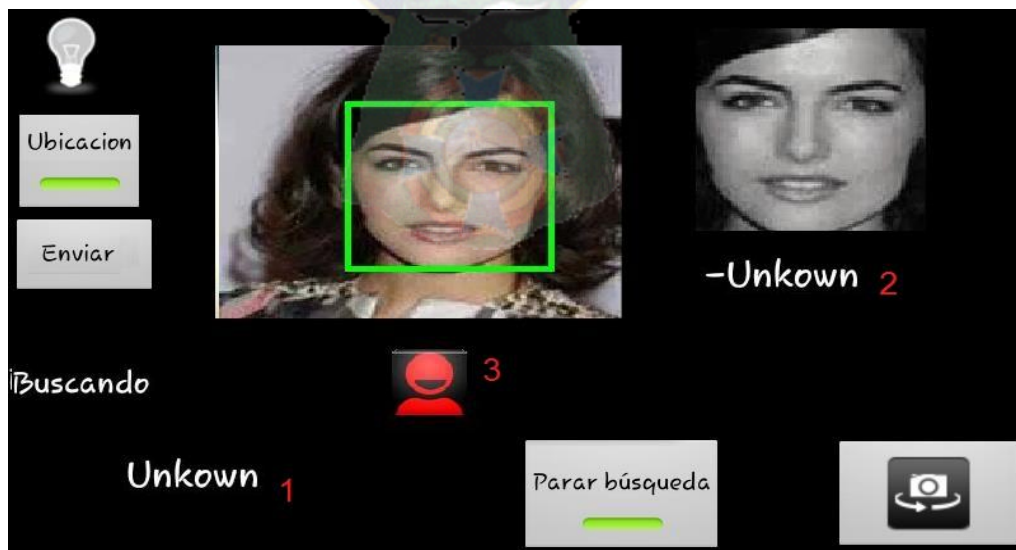


Figura 3.8 Imagen de la aplicación, discriminación Rostro “NO VALIDO” persona no registrada

En la figura 3.9 se muestra el proceso de reconocimiento para un rostro que se encuentra registrado en la base de datos, en el cual tras realizarse el inicio de búsqueda al accionar el botón, nos muestra en tiempo real los datos registrados pertenecientes al rostro como ser el nombre y la descripción del mismo.

En este proceso se muestran los siguientes datos recuperados:

4. Nombre
5. Descripción
6. Icono de reconocimiento semáforo color verde (Valido)

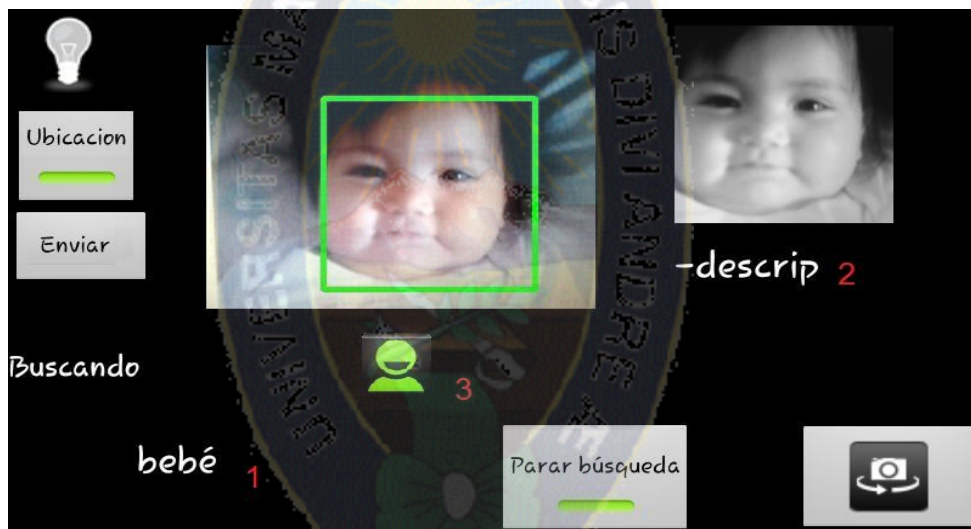


Figura 3.9 Imagen de la aplicación, discriminación Rostro “VALIDO” persona registrada identificada

3.5. Estabilización

En esta fase se realizarán las últimas acciones de integración de la aplicación, quedando el resultado final gráficamente descrito como se ve reflejado en la figura 3.10.



Figura 3.10 Diagrama de actividades: integración de la aplicación

El diagrama de la figura 3.10 representa cada uno de los procesos realizados por actividad en la aplicación. Primero se tiene la activación de la cámara del dispositivo, a través de la cual capturamos las imágenes de la realidad. Luego, la actividad donde se capturan los rostros de las personas escaneadas, que están señalados para activar la realidad aumentada. A continuación, se tiene la actividad en donde se muestra los mensajes correspondientes al procesamiento de los Rostros escaneados, después de su validación.

Por último se tienen las actividades destinadas a la interacción con el usuario, mostrándole mensajes de la aplicación y el botón para el inicio del escaneo de rostros y botón de búsqueda de rostro.

3.6. Pruebas del Sistema

3.6.1. Pruebas en la Aplicación

En principio, en el sistema tenemos dos procesos, los mismos se explican a continuación:

3.6.1.1. Agregación del tipo de personas Buscadas

El sistema en principio debe tener un grupo de personas registradas como desaparecidas para poder realizar el reconocimiento, este es el paso principal para poder poner en funcionamiento del sistema, por lo cual en principio deberán registrarse este grupo/tipo de Elementos haciendo el uso del material requerido para el registro inicial; Se dan dos casos para este registro.

Primer caso

En este caso no se cuentan con las personas desaparecidas de manera física, los detalles de este caso se muestran en la tabla 3.15.

Tabla 3.15 Caso 1 recopilación datos para agregación de personas del tipo “Buscadas”.

Recopilación de Datos	Observaciones
Se hace el Uso de Fotografías las cuales nos brindaran la información requerida del rostro y datos	En este caso deberá existir una persona responsable de brindar las fotografías correspondientes a la persona desaparecida, además deberá otorgar la Información requerida que acompañara como descripción al rostro.

Segundo Caso

En este caso si se cuenta con la persona Físicamente, los detalles y datos necesarios serán proporcionados por el mismo, la recopilación es descrita en la tabla 3.16.

Tabla 3.16 Caso 2, recopilación de datos para agregación de personas del tipo “Buscadas”.

Recopilación de Datos	Observaciones
Se hará el uso del rostro de la persona directamente	En este caso la persona considerada como desaparecida, nos brindara la información requerida por el sistema, esto con el fin de otorgar datos más precisos para su Registro.

El proceso de registro se realiza de la siguiente manera:

- Una persona Responsable/Desparecida hace el uso de la aplicación

- Registra con los elementos y datos solicitados al rostro de persona, la cual será denominada en la base de datos como desaparecida.
- La Persona Responsable/Desaparecida podrá realizar una denuncia oficial con el rostro como fuente principal de información.
- Como Conclusión de este proceso, el Administrador de la Base de datos, se hara cargo de mantener la información un lapso de una semana, mientras se realiza la denuncia oficial, después la cual se podrá mantener los datos y el rostro hasta que se halle a la persona que realiza la búsqueda.

3.6.1.2. Recuperación de datos de Persona Encontrada

El proceso de Escaneo en tiempo real, para el reconocimiento de personas desaparecidas “Encontrada”, se describe en la tabla 3.17.

Tabla 3.17 Recuperación de datos de Persona Encontrada

Recopilación de Datos rostro “Encontrada”	Observaciones
Se hará el uso del rostro de la persona directamente	En este caso la persona considerada como desaparecida, será escaneada para saber si es del tipo “Buscado”, si lo es se tendrá la opción de enviar la imagen de la captura con los datos de ubicación por tecnología GPS con lo que se podrá iniciar el registro de Hallazgo de persona encontrada.

3.6.1.3. Estándares de Material Requerido para el registro y reconocimiento

Se realizaron las pruebas correspondientes en distintos dispositivos móviles para determinar la distancia de escaneo entre el rostro y el dispositivo sin y en movimiento medio, esto según el tamaño del rostro,

Las distancias promedio hasta lograr el enmarcado de rostro reconocido se reflejan en la tabla 3.18.

Tabla 3.18 Pruebas referentes al tamaño rostro escaneo sin movimiento

Tamaño del Rostro escaneado [cm]	Distancia mínima [cm]	Distancia máxima [cm]
Pequeño	12	45
Medio	13	50
Grande	14	52

Respecto al Escaneo de los rostros el escenario ideal es que el rostro pueda ser reconocido y el estándar de rostro se muestra según las siguientes características:

1. Un ambiente claro para tener precisión de detalles del Rostro.
2. Para el primer caso de registro, el Rostro debe evitar realizar Gestos exagerados y rápidos, ya que la información podría truncarse, por lo que se sugiere mantener un rostro calmado evitando muecas.
3. Debe Evitarse el exceso de maquillaje para tener datos más precisos.
4. El rostro debe estar despejado, intentando tener la menor cantidad de elementos encima que puedan volver inexacto la información del reconocimiento.

Estos puntos descritos nos permitirán tener datos precisos a la hora del proceso de escaneo para su mejor precisión.

3.6.2. Funcionamiento de la Aplicación

En este punto se describen los casos de estudio realizados para evaluar el desempeño de la aplicación. Estas pruebas fueron realizadas mediante un estudio de casos en situaciones límites, ideales y normales a las que la aplicación pueda ser expuesta.

En la tabla 3.19 se describen los dispositivos móviles en los cuáles las pruebas fueron llevadas a cabo y seguidamente describimos los casos de prueba.

Tabla 3.19 Dispositivos utilizados para las pruebas.

Dispositivo	Versión de Android	Resolución Pantalla	Densidad
Samsung Galaxy Ace	2.3.6	320 x 480	2592x1944 pixeles

Samsung Galaxy Ace Plus	2.3.6	320 x 480	2592x1944 pixeles
Galaxy Nexus	4.0	720 x 1280	2592x1936 pixeles
Samsung S3 mini	4.02	800x480	233 ppi por pixeles
Sony Xperia J2	5.1.1	540 x 960	234 ppp por pixeles

A continuación se listan los casos de estudios realizados para la prueba de la aplicación

3.6.2.1. Caso de estudio 1

Descripción. En este caso de estudio se tomó una situación ideal en la que el escaneo de la persona se realice con las condiciones climáticas de estándares de reconocimiento y acceso a red, para los cuales describimos los mismos en la siguiente lista.

- Ambiente claro
- Usuario con acceso a Internet
- El rostro escaneado se encuentra en los estándares de reconocimiento

Resultados. La aplicación cumple los objetivos y requerimientos de funcionalidad.

Observaciones. En este caso de estudio no se tiene ninguna observación

3.6.2.2. Caso de estudio 2

Descripción. En este caso de estudio se tomó una situación en la que el escaneo de la persona se realice con las condiciones climáticas no ideales, pero si las de estándares de reconocimiento y acceso a la red, para los cuales describimos los mismos en la siguiente lista.

- Ambiente obscuro
- Usuario con acceso a Internet
- El rostro escaneado se encuentra en los estándares de reconocimiento

Resultados. La aplicación cuenta con la opción de linterna, por lo cual al momento de realizar el escaneo, cumple los objetivos y requerimientos de funcionalidad.

Observaciones. En este caso de estudio no se tiene ninguna observación

3.6.2.3. Caso de estudio 3

Descripción. En este caso de estudio se tomó una situación en la que el escaneo de la persona se realice con las condiciones climáticas y de acceso ideales, pero no así las de estándares de reconocimiento, para los cuales describimos los mismos en la siguiente lista.

- Ambiente claro
- Usuario con acceso a Internet
- El rostro escaneado no se encuentra en los estándares de reconocimiento.

Resultados. La aplicación, no reconoce de manera efectiva los rostros escaneados, por lo cual al momento de realizar el reconocimiento se frustra o ingresa datos erróneos, se concluye que esto puede causar que la aplicación no cumpla los objetivos y requerimientos de funcionalidad.

Observaciones. En este caso de estudio se demostró que se requiere limitarse a de los estándares de reconocimiento para tener un mejor procesamiento de datos.

3.6.2.4. Caso de estudio 4

Descripción. En este caso de estudio se tomó una situación en la que el escaneo de la persona se realice con las condiciones climáticas y de acceso ideales, pero no así las de estándares de reconocimiento, para los cuales describimos los mismos en la siguiente lista.

- Ambiente claro
- Usuario con acceso a Internet
- El rostro escaneado no se encuentra en bastante movimiento.

Resultados. La aplicación cuenta con un proceso de grabado ya que está dentro de los márgenes de realidad aumentada, pero al no poder enfocar el rostro de manera completa y efectiva, cae en errores de autenticación o se obtiene una imagen no clara, la aplicación tiene la opción de hacer otras capturas más, por lo que el rostro fue captado de modo exitoso gracias

al método grabación y estandarización del rostro, por lo que concluimos que la aplicación cumple los objetivos y requerimientos de funcionalidad pero de manera menos efectiva.

Observaciones. En este caso de estudio se demostró que la aplicación requiere un ambiente quieto para su registro, o con muy poco movimiento

3.6.2.5. Caso de estudio 5

Descripción. En este caso de estudio se tomó una situación en la que el escaneo de la persona se realice con las condiciones climáticas y de estándares de reconocimiento ideales, pero no así las de acceso, para los cuales describimos los mismos en la siguiente lista.

- Ambiente claro
- Usuario sin acceso a Internet
- El rostro escaneado se encuentra en mucho movimiento

Resultados. La aplicación requiere de acceso a internet, esto para poder tener realizar el registro o hallazgo, no siendo así para reconocer, por lo cual si se carece de este, se requerirá de otro procedimiento de pausa de registro hasta hallar el acceso de internet requerido.

Observaciones. En este caso se observa el uso Necesario del Acceso a Internet, si no el registro de búsqueda o hallazgo no se podría realizar en tiempo real.

ANALISIS DE DATOS Y RESULTADOS

4.1. Prueba de Hipótesis

En esta sección se procederá a la realización de la evaluación de la hipótesis planteada en el capítulo primero.

“El Reconocimiento facial aplicando realidad aumentada en aplicaciones móviles identifica, ubica a las personas de manera fácil y pronta en casos de personas desaparecidas”.

De este proceso identificamos las variables independiente, dependiente e interviniente:

- Variable Independiente: aplicaciones móviles
- Variable dependiente: identifica de manera fácil, ubica de manera pronta
- Variable Interviniente: Reconocimiento facial, realidad Aumentada

4.2. Contraste de Rachas de Wald – Wolfowitz

La prueba de rachas también denominada “Prueba de Wald - Wolfowitz” después de Abraham Wald y Jacob Wolfowitz, es una no paramétrica prueba de estadística la cual comprueba la hipótesis aleatoria de una secuencia de datos de dos valores.

Se supone una población cuya función de distribución es desconocida y sea X la variable Aleatoria asociada a esa población la cual puede tomar dos posibles valores como por ejemplo, éxito (A) y fracaso (B):

Dónde:

H_0 : La muestra es aleatoria

H_1 : La muestra no es aleatoria

En general, sea una muestra de tamaño n en la que han aparecido n_1 elementos del tipo A y n_2 elementos del tipo B

- Llamaremos R_{1j} al número de rachas del elemento **1(A)** de longitud j
- Llamaremos n_1 , al número d rachas del elemento **1(A)** en la muestra
- Llamaremos R_{2j} al número de rachas del elemento **2(A)** de longitud j
- Llamaremos n_2 , al número d rachas del elemento **2(A)** en la muestra
- El total de la muestra será: $n = n_1 + n_2$
- El total de rachas será: $R_{exp} = R_{1j} + R_{2j}$, el número de rachasen la muestra

Bajo la hipótesis H_0 es decir, para muestras aleatorias la distribución de probabilidad de R tiende hacia la normal a medida que n_1 y n_2 se van haciendo grandes.

Esta aproximación es óptima si $n_1 > 10$ y $n_2 > 10$; de tal manera que la esperanza y varianza serán resueltos con las ecuaciones que se indican en la tabla 4.1.

Tabla 4.1 Esperanza y Varianza

$R_{exp} \rightarrow (E R_{exp}, Var[R_{exp}])$	
Esperanza	$E[R_{exp}] = \frac{2 * n_1 * n_2 + n}{n}$
Varianza	$Var[R_{exp}] = \sigma[R_{exp}] = \frac{2 * n_1 * n_2 (2 * n_1 * n_2 - n)}{n^2 * (n - 1)}$

Por consiguiente para muestras grandes se verifica:

$$Z = \frac{R - E[R_{exp}]}{\sigma[R_{exp}]}$$

Y para una muestra concreta el valor del estadístico Z será:

$$Z_{exp} = \frac{R - \frac{2 * n_1 * n_2 + n}{n} - 0.5}{\frac{2 * n_1 * n_2 (2 * n_1 * n_2 - n)}{n^2 * (n - 1)}}$$

Para la prueba, nos planteamos una encuesta de valoración del servicio móvil para la prevención de criminalidad urbana, la cual estudia los usuarios que necesitan de una aplicación móvil de esta categoría y su aceptación sobre el software como servicio además, la evaluación de las variables involucradas en la hipótesis.

De los resultados de la encuesta, con los valores tomados a mayor rango del 50% de aceptación o rechazo, se obtiene la tabla 4.2, más detalles en el anexo 7:

Tabla 4.2 Propuesta de software como servicio y rango de aceptación del usuario

	Preguntas	Software como servicio	Aceptación por rachas
1	Estructura general del servicio móvil	Mejora	+
2	Acceso a la aplicación móvil	No mejora	-
3	Registro de nombre	No mejora	-
4	Registro de datos de descripción	Mejora	+
5	Registro de rostro de persona desaparecida	Mejora	+
6	Registro de denuncia	Mejora	+
7	Visualización de nombre al escaneo de rostro	No mejora	-
8	Visualización de datos e información al escaneo de rostro	Mejora	+
9	Opciones de localización	Mejora	+
10	Método de identificación(escaneo)	Mejora	+
11	Registro de Identificación	Mejora	+
12	Tiempo de Registros	Mejora	+
13	Tiempo de identificación	Mejora	+
14	Intervalo de tiempo entre registro e Identificación (lapso de búsqueda).	Mejora	+

Se obtuvieron los siguientes resultados:

(+)(--)(+++)(-)(+++++++)

Dónde:

- (-) Representa los casos en los que el usuario no muestra aprobación o una aprobación mínima en la cual no indica mejora.
- (+) Representa casos en los que el servicio móvil mostro aprobación por parte del usuario en comparación con el método actual de búsqueda y ubicación de personas.

Siendo una racha construida por la sucesión de signos iguales se tiene:

Total de Rachas expuestas: $R_{exp}=5$

Número total de observaciones: $N=14$

Numero de Residuos Positivos: $n_1 = 11$

Numero de residuos negativos: $n_2 =3$

Reemplazando datos para calcular la Esperanza y Varianza se tiene:

Esperanza:

$$E [R_{exp}] = \frac{2 * n_1 * n_2 + n}{n} = \frac{2 * 11 * 3 + 14}{14} = 5.714$$

Varianza:

$$\begin{aligned} \text{Var} [R_{exp}] &= \sigma [R_{exp}] = \frac{2 * n_1 * n_2}{n^2 * n - 1} = \\ &= \frac{2 * 11 * 3}{14^2 * 14 - 1} = 1.347 \end{aligned}$$

4.3. Toma de Decisión

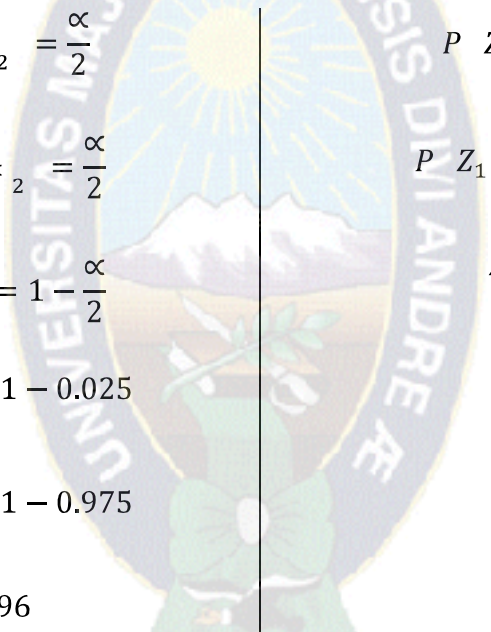
Y para la muestra concreta el valor del estadístico Z_{exp} reemplazando datos se tiene:

$$Z_{exp} = \frac{R_{exp} - E R_{exp} - 0.5}{\sigma[R_{exp}]} = \frac{5 - 5.714 - 0.5}{1.347} = 0.188715664$$

Para calcular la región de aceptación de la hipótesis, es necesario hallar el valor de $Z_{\alpha/2}$ que se obtiene de la tabla de la Normal N (0,1), de manera que cumple:

$$P Z_1 \leq -Z_{\alpha/2} = P Z_1 \geq Z_{\alpha/2} = \frac{\alpha}{2}$$

Al 95% de nivel de confianza



$$\begin{aligned}
 P Z_1 \leq -Z_{\alpha/2} &= \frac{\alpha}{2} & P Z_1 \geq Z_{\alpha/2} &= \frac{\alpha}{2} \\
 1 - P Z_1 < Z_{\alpha/2} &= \frac{\alpha}{2} & P Z_1 \geq Z_{\alpha/2} &= 0.025 \\
 P Z_1 < Z_{\alpha/2} &= 1 - \frac{\alpha}{2} & Z_{\alpha/2} &= 1.96 \\
 P Z_1 < Z_{\alpha/2} &= 1 - 0.025 \\
 P Z_1 < Z_{\alpha/2} &= 1 - 0.975 \\
 Z_{\alpha/2} &= 1.96
 \end{aligned}$$

Por lo tanto la región de la aceptación de la hipótesis nula es:

$$-Z_{\alpha/2} < Z_{exp} < Z_{\alpha/2}$$

$$-1.96 < 0.159 < 1.96$$

Se puede ver que el estadístico $Z_{exp} = 0.159$ se encuentra en el intervalo de la hipótesis, por lo tanto se puede afirmar la variable dependiente Identifica de manera fácil, ubica de manera pronta.

Lo que demuestra que la tesis es un trabajo valido, además indica que los datos de la muestra son aleatorios.



CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Se ha desarrollado el modelo con las metodologías aplicadas con lo que se cumplió el objetivo general fue cumplido ya que se logró la construcción de una aplicación utilizando realidad aumentada la cual reconoce, valida e identifica Rostros de personas en tiempo real. Además, el usuario logra obtener información en base a la misma además de la ubicación logrando realizarse las

- Se examinó los distintos SDK´s de realidad aumentada para la plataforma Android y se tomó en cuenta sus características para la elección final, el cual logra hacer el uso de Ubicación y la recuperación del mismo.
- Se logró el procesamiento de la información almacenada en los Rostros escaneados, mediante el uso del SDK y la librería OpenCV. También se logró que el usuario tuviera acceso a la información con el escaneo de reconocimiento.
- Se procesó la información recuperada al escaneo del rostro con la tecnología de realidad aumentada y su envió en el proceso de registro.
- Se diseñó e implementó la base de datos que almacena la información de los rostros y su tipo. En el servidor que se encuentra realizado en el sistema de gestión de Base de Datos *MySQL*₁.
- La construcción modelado y diseño de una aplicación con realidad aumentada según resultados en encuestas reduce el tiempo y espacio entre la ubicación de personas que son buscadas y personas que fueron encontradas, además resulta ser una nueva utilidad en el uso de dispositivos Smartphone.
- Si un dispositivo no cuenta con el acceso a Internet, un usuario no puede obtener los datos e información del rostro, además de no poder registrarlo en la base de datos, por lo que el acceso a la red es un elemento imprescindible para la herramienta en su modo de registro.

- El usuario tiene la opción de realizar el registro de una persona desaparecida y una persona encontrada al administrador final de la herramienta.
- Se implementaron distintas pruebas sobre la aplicación, las cuales mostraron distintos resultados favorables tomando en cuenta diversos factores y casos de uso.
- Puesto que la herramienta de reconocimiento ubica reconoce y brinda información a partir del rostro, puede utilizarse en otros ámbitos donde esta herramienta pueda ser requerida.
- La población si utilizaría una aplicación orientada a la seguridad ciudadana, además que es muy probable que recomiende su uso según datos obtenidos en las encuestas.

5.2. Recomendaciones

- Se recomienda que el rostro siga los estándares para el registro y reconocimiento, en ambiente claro y demás características descritas en el capítulo de estabilización, esto para lograr que los datos recopilados y recuperados, sean lo más exactos posibles para su mejor utilización.
- Para una mejor verificación de la validez del rostro, la aplicación y desarrollo de más librerías y algoritmos de reconocimiento esto para optimizar las actividades de la realidad aumentada para el reconocimiento de los rostros y mayor exactitud en la misma.
- Se recomienda llevar un registro actualizado de los rostros en cuanto a la eliminación de rostros cuando logren ser resueltos en el medio en el que sean empleados, esto para una mejor administración de los datos.
- Construir otras aplicaciones orientadas a la seguridad, a la reducción de tiempo y eficiencia de ubicación con el uso de la realidad aumentada como por ejemplo, ubicación de servicios emergencia, registros en tiempo real, etc.

BIBLIOGRAFÍA

- Aguerreberere C. & Capdehourat G. (2015). *Reconocimiento de caras con características locales*. (Proyecto Final Reconocimiento de Patrones). Instituto de Ingeniería Eléctrica, Universidad de la Republica.
- Ahonen T., Hadid A., & Pietikainen M. (2004). Face recognition with local binary patterns. *Machine Vision Group*, in the 8th European Conference on Computer Vision. (1):469-481. University of Oulu, Finland. Infotech Oulu. Recuperado de [http :
==www:cs:colostate:edu=evalfacerec=papers=EBGMThesis Final:pdf](http://www.cs.colostate.edu/~evalfacerec/papers/EBGMThesisFinal.pdf)
- Alvarado S. (2013). *Realidad Aumentada en Aplicaciones Móviles para la Seguridad Ciudadana en el Transporte Público*. La Paz. Bolivia
- Blanco, P., Camarero, J., Fumero, A., Werterski, A., & Rodriguez, P. (2009). *Metodología de desarrollo ágil para sistemas móviles*. Universidad Politécnica de Madrid. Madrid.
- Bolivia Informa. (2011). *Bolivia Informa*. Recuperado de: <http://reyquibolivia.blogspot.com/2011/10/la-paz-el-75-reprueba-al-transporte.html>.
- Bolme, Beveridge J.R., Teixeira M., & Draper B.A. (2004). *Evaluation of face recognition algorithms*. D.S. Recuperado de [http://www:cs:colostate:edu=evalfacerec=algorithms5.html](http://www.cs.colostate.edu/~evalfacerec/algorithms5.html)
- Bonnin, J. (2008). *Realidad Aumentada*. Recuperado de <http://outernet.tk/>
- Clemments , P., & Northrop, L. (2003). *Software Product Lines, course notes of Product Line Systems Program*. Carnegie Mellon University: Software Engineering Institute.
- Ciencia y Computación (2014) El Diario *periodico nacional*.La Paz Bolivia.
- Cuera, M. E. (2013). *Software Quality Assurance*. Recuperado el 21 de 08 de 2015, de <http://www.mtp.es/index.php/sala-de-prensa/articulos-de-opinion?id=265>
- EABolivia (2016). *Seguridad: FELCC registró 180 personas extraviadas en la ciudad de La Paz desde enero*. La Paz Bolivia Recuperado de <http://www.eabolivia.com/seguridad/13825-felcc-registro-180-personas-extraviadas-en-la-paz-desde-enero.html>
- Fundación Telefónica. (2011). *Realidad aumentada: una nueva lente para ver el mundo*. Ariel S.A. España. Recuperado de <http://www.realidadaugmentada-fundaciontelefonica.com/realidad-aumentada.pdf>
- García J. y Príncipe L. A. (2014). *Desarrollo de un Sistema Móvil como Apoyo a las Comisarias en la Seguridad Ciudadana de la Ciudad de Trujillo* (Trabajo de

Graduacion) Universidad Nacional de Trujillo.Trujillo. Perú. Recuperado de <http://www.inf.unitru.edu.pe/revistas/2014/2.pdf>

Gonzales T. (2013). *Análisis, Diseño e Implementación de un sistema web y móvil para el soporte informático a la gestión de los servicios de atención que brindan las comisarías a la comunidad.* (Tesis de Grado) Pontifica Universidad Católica del Perú. Lima Perú Recuperado de <http://tesis.pucp.edu.pe/repositorio/handle/123456789/4848>

Gobierno Municipal de La Paz. (2013). *Periodico del Gobierno Municipal de La Paz.* . La Paz: Gobierno Municipal de La Paz. Recuperado de: http://www.lapaz.bo/index.php?option=com_content&view=article&id=4814&Itemid=541

Gobierno Municipal Nuestra Señora de La Paz. (2009). *Reglamento Municipal para el Funcionamiento de Radio Taxis en el Municipio de La Paz.* La Paz: Gobierno Municipal de La Paz.

Georghiadis A.S., Belhumeur P.N., & Kriegman D.J. (2001). *From few to many: Illumination cone models for face recognition under variable lighting and pose.* Pattern Anal. Mach. Intelligence, 23(6), 643–660.

González Gutiérrez, D., Remis García, S., & Rubén Fernández, S. (2011). *Realidad Aumentada.* Universidad de Oviedo: Escuela Politécnica de Ingeniería de Gijón.

Hyunjong Ch. & Seungbin M.(2009). *Comparison of pca and lda based face recognition algorithms under illumination variations.*

Instituto Nacional de Estadística. (2012). *Censo nacional de población y vivienda.* La Paz Recuperado de <http://www.ine.gob.bo:8081/censo2012/PDF/resultadosCPV2012.pdf>

Kak A., & Martinez. A. (2001). *Pca versus lda.* IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. 23(2). Recuperado de <http://www2.ece.ohio-state.edu/~aleix/pami01.pdf>

Lee K.C., Ho J., & Kriegman D. (2005). Acquiring linear subspaces for face recognition under variable lighting: *Trans. Pattern Anal. Intelligence*, 27(5):684–698.

Observatorio Nacional de Seguridad Ciudadana de Bolivia. (05 de 03 de 2013). *OBSERVATORIO NACIONAL DE SEGURIDAD CIUDADANA DE BOLIVIA.* Recuperado de <http://www.onsc.gob.bo/>.

OpenCV. (2014). *OpenCV docs.* Recuperado de: http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_api.html

- OpenCV. (2014). *Facerecognizer*. Recuperado de: http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_api.html#FaceRecognizer:publicAlgorithm.
- Ottado G. (2015). *Reconocimiento de caras: Eigenfaces y Fisherfaces*. Recuperado de https://eva.fing.edu.uy/file.php/514/ARCHIVO/2010/TrabajosFinales2010/informe_final_ottado.pdf.
- Pentland A., Etcoff N., Masoiu A., Oliyide O., Starner T., & Turk M. (1993). Experiments. *eigenfaces*. In *Looking At People Workshop*, IJCAI.
- Pereyra P.A. (2015) Reconocimiento Facial Mediante Imágenes Estereoscópicas Para Control de Ingreso (Tesis de Grado) Universidad de Buenos Aires. Argentina Recuperado de <http://materias.fi.uba.ar/7500/PamelaPereira.pdf>
- Periódico El Cambio. (31 de 01 de 2013). *PERIODICO EL CAMBIO*. Recuperado el http://www.cambio.bo/ciudades/20130131/se_controlara_a_los_radiotaxis_con_gps_87966.htm
- Periódico El Cambio. (31 de 01 de 2013). *Noticias*. recuperado de http://www.cambio.bo/ciudades/20130131/se_controlara_a_los_radiotaxis_con_gps_87966.htm
- Periódico Página Siete. (19 de 05 de 2012). *Destacados*. Recuperado de <http://www.paginasiete.bo/2012-09-05/Nacional/Destacados/34Seg01050912.aspx>
- Periódico Página Siete. (10 de 09 de 2012). *Destacados*. Recuperado de <http://www.paginasiete.bo/2012-09-10/Sociedad/Destacados/340AdeSoc010912lun10.aspx>
- Periódico El Día. (2015). *Trata y tráfico*. Recuperado de https://www.eldia.com.bo/mobile.php?cat=1&pla=7&id_articulo=62315.
- Peter N. Belhumeur, Joao P. Hespanha, & David J. Kriegman. (1996). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. En Bernard F. Buxton and Roberto Cipolla, (ED), ECCV (1), *volume 1064 of Lecture Notes in Computer Science*, 45–58. Springer, Recuperado de <http://www.ine.gov.bo:8081/censo2012/PDF/resultadosCPV2012.pdf>
- Rahimian, V., & Ramsin, R. (2008). Designing and agile methodology for mobile software. En S. I. Science.
- Red. (2013). *Red*. Recuperado de <http://www.red.es/redes/sala-de-prensa/nota-de-prensa/las-aplicaciones-moviles-como-llave-de-acceso-del-ciudadano-la-ciudad->

Rodríguez T. (2011). *genbetadev*. Recuperado de <http://www.genbetadev.com/desarrollo-aplicaciones-moviles/metodos-aplicables-para-el-desarrollo-de-aplicaciones-moviles>.

Ruiz-del-Solar J. & Navarrete P. (2002). Analysis and comparison of eigenspace-based face recognition approaches. *International Journal of Pattern Recognition and Artificial Intelligence*.

Tribuna abierta. (2011). *Tribuna abierta*. Recuperado de: <http://www.tribunaabierta.com/blog/?p=314>.

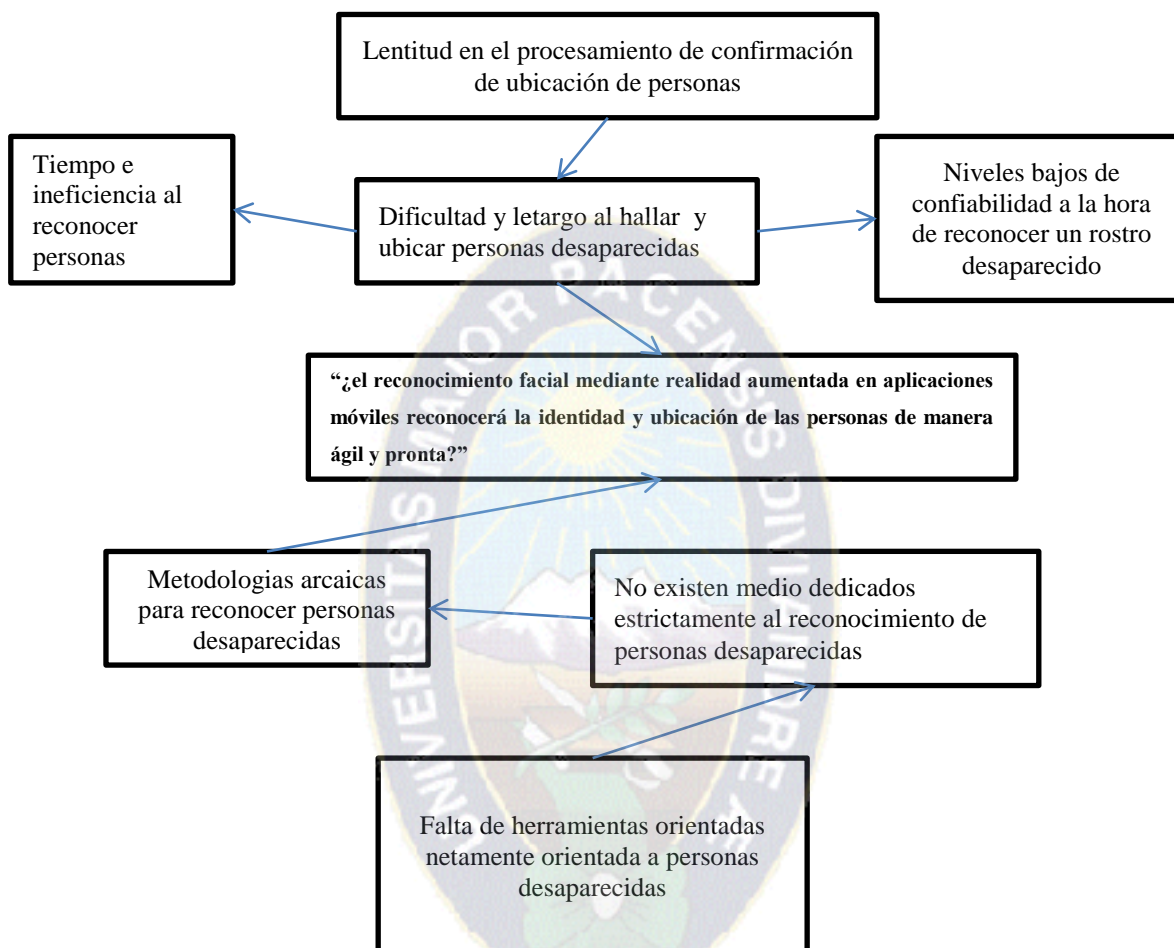
Vargas R. (2012). *Sistema de alerta basado en tecnología móvil*. La Paz. Bolivia

Zúñiga F. (2015) Reconocimiento frontal de rostros en base a imágenes de alta resolución
Recuperado de <http://repositorio.uchile.cl/bitstream/handle/2250/134095/Reconocimiento-frontal-de-rostros-en-base-a-imagenes-de-alta-resolucion.pdf?sequence=1>

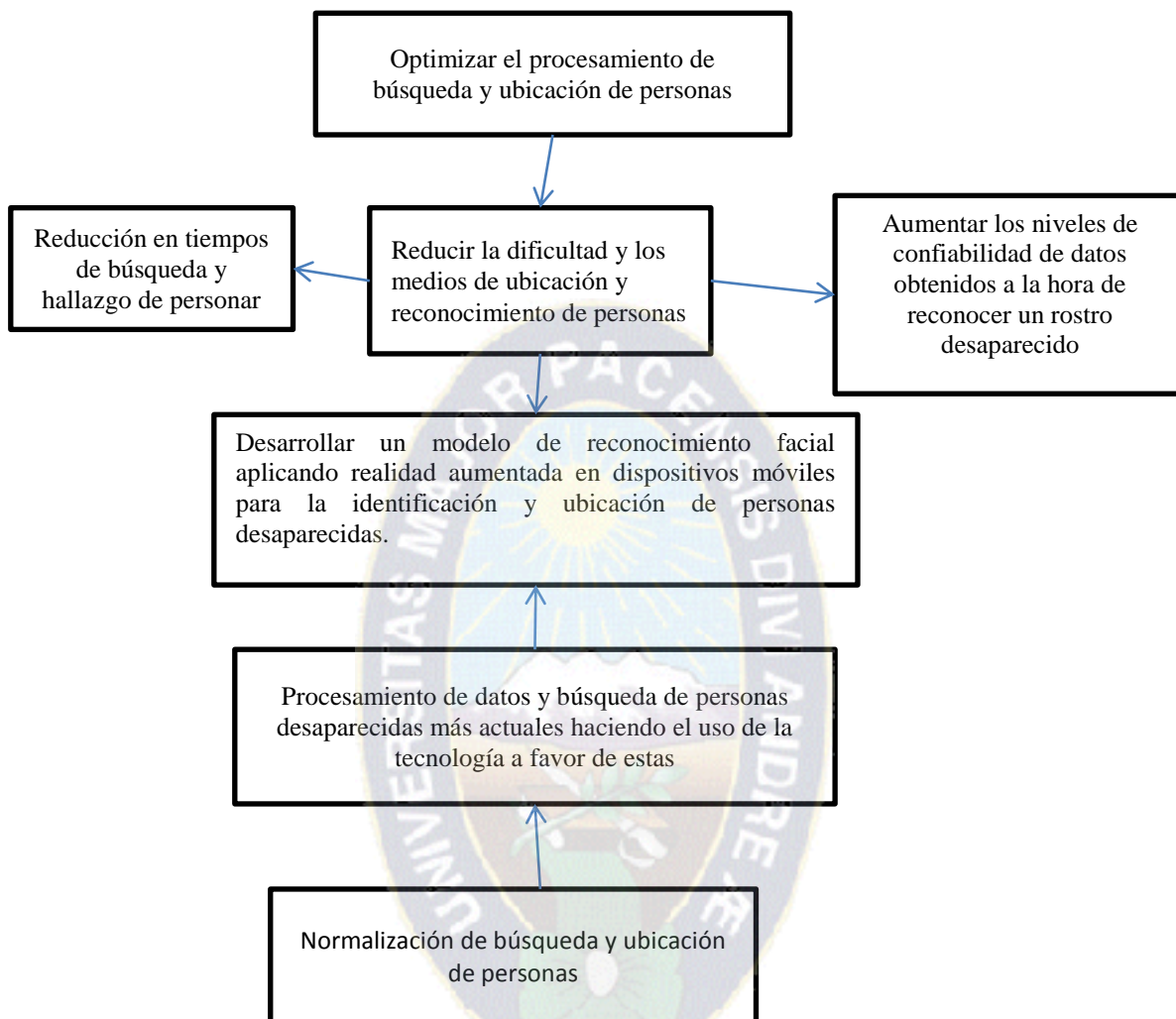


ANEXOS

Anexo A. Árbol de problemas



Anexo B. Árbol de objetivos



Anexo C. Marco Lógico

Resumen Narrativo	Indicadores	Medios de Verificación	Supuestos
<p>Fin</p> <p>La presente tesis colaborara a mejorar y reducir el tiempo de ubicación de personas</p>	<p>Impactos</p> <p>Detección de la información extraída de la base de datos</p> <p>Reducción de tiempo al ubicar y denunciar un caso de persona a encontrar.</p>	<p>Medios de Verificación.</p> <p>Las pruebas que se realicen serán 1 analítica propuesta, ocasionando un análisis estadístico</p>	<p>Sostenibilidad</p> <p>Nuevas formas de procesamiento de datos</p>
<p>Propósito</p> <p>Reducción de tiempo</p>	<p>Resultados</p> <p>Pruebas de muestreo que coadyuven a demostrar la hipótesis.</p>	<p>Medios de Verificación</p> <p>Análisis estadísticos de las pruebas.</p>	<p>Propósito a Fin</p> <p>Reconocimiento específico eficiente y confiable de un rostro obtenido</p>
<p>Componentes/Productos</p> <p>Análítica predictiva para mejorar la eficiencia de hallazgo de personas</p>	<p>Productos</p> <p>Se esperara una eficiencia de un 95 % de confianza en un plazo no mayor de 3 meses</p>	<p>Medios de Verificación</p> <p>Se realizaran pruebas de muestro que permitirán demostrar la confiabilidad de la analítica.</p>	<p>Componentes/ Productos a Propósito</p> <p>Condiciones para la analítica es contar con un servidor y una base de datos contenga la información necesaria para el manejo de la misma</p>
<p>Actividades</p> <p>Análisis de datos</p> <p>Análisis de la Base de Datos</p> <p>Vulnerabilidades</p> <p>Inseguridad</p> <p>Informe de pruebas</p>	<p>Análisis de los datos y el tiempo de la investigación 2 semanas.</p> <p>Análisis de la Base de Datos Relacional y el tiempo de investigación 1 semana.</p> <p>Vulnerabilidades tiempo de investigación 1</p>	<p>Medios de Verificación</p> <p>Encuestas, reportes</p>	<p>Actividades a Componentes</p> <p>Condiciones de seguridad para mejorar la analítica.</p> <p>Estabilidad de datos</p>

	semanas. Inseguridad tiempo de investigación 1 semanas. Informe de pruebas tiempo de investigación 1 semanas.		
--	---	--	--



Anexo D: Desarrollo de la aplicación

En este apéndice se describe a continuación la totalidad de tablas y figuras de las distintas fases de la metodología Mobile-D, seguida para el desarrollo de la aplicación.

Historias de usuario y tareas

En la tabla A.1 se describe la tarea de ingeniería para el diseño de la base de datos.

Tabla A.1 Tarea de ingeniería: Diseño de base de datos

Tarea de Ingeniería	
Número Tarea: 3.1	Número Historia de Usuario: 3
Nombre Tarea: Diseño e implementación de la base de datos	
Tipo de Tarea : Diseño - Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Programador Responsable: Libia Gonzales	
Descripción: Diseñar una base de datos para almacenar la información referente a los rostros con los campos de nombre y descripción además del detalle de tipo de persona en la cual se diferenciaron por “Buscado” y “Hallado”	

ANEXO E: Producción

A continuación se describe la totalidad de las iteraciones de esta fase, salvo la iteración 1,3 y 4 los cuales pueden ser referenciados en el Capítulo 3.

E.1. Iteración 2. Integración con el SDK layar

[Historia de usuario 1, tarea de ingeniería 1.2]. La segunda iteración describe la integración de la aplicación con el SDK layar.

E.1.1. Planeación

La fase de planeación de la iteración se describe a continuación en tabla A.2.

Tabla A.2 Planeación iteración 2

CARACTERISTICAS	DURACIÓN
Se integró el SDK de Layar con la aplicación y se probó la funcionalidad de la realidad aumentada. Se comprobó el escaneo y procesamiento brindado por el SDK con reconocimiento de Rostros	3 días

E.1.2. Codificación

La librería “layar_vision_sdk_7_2_3_RC_20130628” fue integrada al proyecto, se probaron sus características y funciones.

E.1.3. Liberación

La fase de liberación de la iteración se describe a continuación en tabla A.3.

Tabla A.3 Caso de prueba iteración 2

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Número Historia de Usuario: 1
Descripción de la Prueba:	

Prueba de integración del proyecto con el SDK layar, prueba de funciones y características
<p>Condiciones de Ejecución:</p> <p>La aplicación deberá abrir una nueva actividad Android, la cual estará integrará el SDK, se probará la activación de la realidad aumentada en cuanto a la opción de ubicación que este proporciona</p>
<p>Entrada / Pasos de ejecución:</p> <p>Escaneo un rostro para su registro y su reconocimiento y ubicación, se inició la actividad integrada con el SDK y se observó el resultado.</p>
<p>Resultado Esperado:</p> <p>Reconocimiento del Rostro como activador de la realidad aumentada.</p>
<p>Evaluación de la Prueba:</p> <p>La prueba fue exitosa demostrando que el sdk fue integrado exitosamente y se encuentra listo para su uso.</p>

E.2. Iteración 5. Diseño e implementación de la base de datos

[Historia de usuario 3, tarea de ingeniería 3.1]. La quinta iteración está orientada a la construcción de la base de datos que almacenará la información de las personas desaparecidas, el tipo, y las descripciones. Para su almacenamiento, diseño e implementación se utilizaron los siguientes recursos.

- Almacenamiento en el servidor WAMP.
- El sistema de base datos elegido fue MySQL.
- La capacidad total de almacenamiento es de 4GB.

En la Figura A.1 se puede apreciar el modelo de la base de datos para el sistema modelado en UML donde se describen las clases Persona, Imagen, PBuscada y PEncontrada con sus respectivos atributos además de la cardinalidad existente entre estos, seguidamente se describen cada una de estas tablas en detalle.

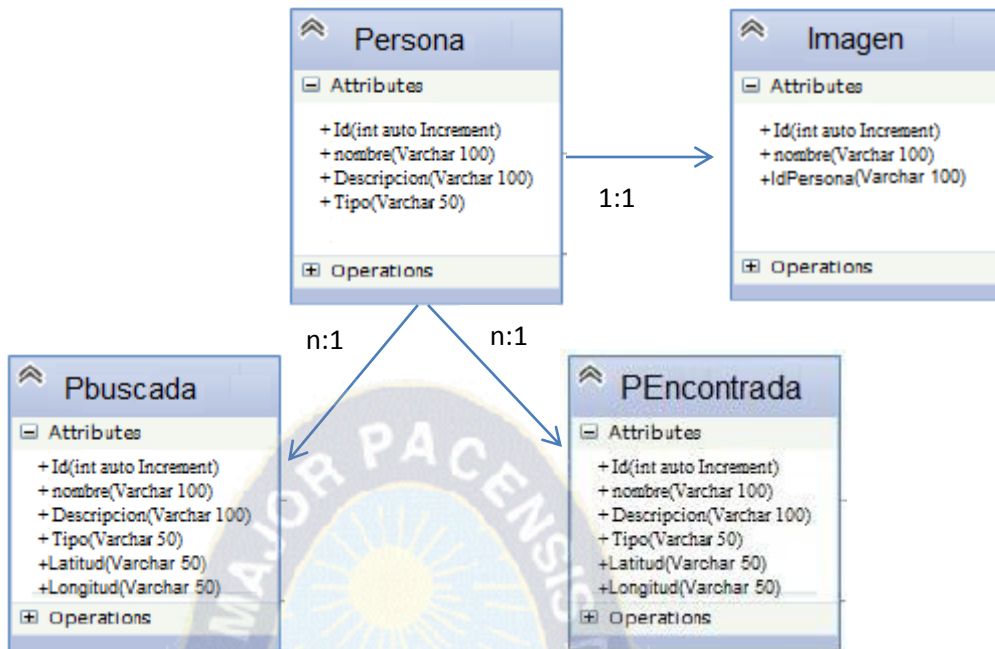


Figura A.1 Modelo de datos

Descripción del modelo de datos. La figura A.1 Muestra el diagrama relacional de las clases, las cuales están relacionadas de una a una con la clase Persona

A continuación detallamos las tablas:

Persona: esta clase almacena los datos principales de la persona en el sistema, sus atributos son los siguientes:

Id: (llave principal, int, not null) Identificador único de cada persona en la tabla.

Nombre: (Varchar 100, not null) Almacena del nombre de la persona.

PBuscada: Esta clase almacena los datos de las personas que son del tipo Buscado

Id: (llave principal, int, not null) Identificador único de cada persona en la tabla.

Nombre: (llave Foranea, Varchar 100, not null) Almacena del nombre de la persona.

Descripción: (varchar 100, not null) Almacena los datos de descripción de la persona

Tipo: (Varchar 50, not null) Almacena el tipo de persona clase a la que pertenece constante “Buscada”

Latitud: (Varchar 50, not null) Almacena el dato de ubicación geográfica latitud

Longitud: (Varchar 50, not null) Almacena el dato de ubicación geográfica longitud

Idp: (Int 100) Llave foránea de la tabla Persona

PEncontrada: Esta clase almacena los datos de las personas que son del tipo Encontrado

Id: (llave principal, int, not null) Identificador único de cada persona en la tabla.

Nombre: (llave Foranea, Varchar 50, not null) Almacena del nombre de la persona.

Descripción: (varchar 100, not null) Almacena los datos de descripción de la persona

Tipo: (Varchar 50, not null) Almacena el tipo de persona clase a la que pertenece constante “Encontrada”

Latitud: (Varchar 50, not null) Almacena el dato de ubicación geográfica latitud

Longitud: (Varchar 50, not null) Almacena el dato de ubicación geográfica longitud

Idp: (Int 100) Llave foránea de la tabla Persona

Imagen: Esta clase almacena los datos de la imagen que será almacenada que esta relacionada con una carpeta en el servidor

Id: (llave principal, int, not null) Identificador único de cada persona en la tabla.

Nombre: (Varchar 50, not null) Almacena del nombre de la imagen la cual se almacenara en una carpeta del servidor.

Idp: (llave foránea int, not null) Almacena el dato de id dela persona a la cual corresponde la imagen.

E.3. Iteración 6. Navegación a través del Rostro al ser Escaneado

[Historia de usuario 3, tarea de ingeniería 3.3]. La séptima iteración describe la por tipos de rostros registrados válidos, los cuales están diferenciados por dos tipos “Buscados” y “Hallados”.

E.3.1. Planificación

La fase de planificación de la iteración se describe a continuación en tabla A.4.

Tabla A.4 Planeación iteración 6

CARACTERÍSTICAS	DURACIÓN
El usuario podrá observar en la pantalla el icono de validez de porcentaje de reconocimiento para los rostros. Esto se muestra al momento del escaneo indicando el porcentaje de semejanza con colores que indican el grado de similitud.	3 días

E.3.2. Codificación

Para esta iteración entre la base de datos y los datos de la aplicación, se hizo uso de un web service a través de las librerías HttpClient Clase encargada de enviar la información almacenada en httpPost alWebService. HttpPost: almacena los datos que serán enviados, por medio de HttpClient, al WebService, en el cual se enviara nuestros datos almacenados en listas las cuales serán recibidas por el Servidor.

E.3.3. Liberación

La fase de liberación de la iteración se describe a continuación en tabla A.5

Tabla A.5 Caso de prueba iteración 6

Caso de Prueba de Aceptación	
Código Caso de Prueba: 4	Número Historia de Usuario: 3
Descripción de la Prueba: Funciones de navegación en la aplicación a través de un Rostro escaneado válido	

<p>Condiciones de Ejecución:</p> <p>El Rostro a través del que se intenta navegar deberá estar en modo “Valido” indicado por las imágenes de similitud de rostros en semáforo [iteración 4]</p>
<p>Entrada / Pasos de ejecución:</p> <p>La entrada debe ser un Rostro escaneado válido</p>
<p>Resultado Esperado:</p> <p>El rostro válido al ser encontrado otorgara los datos del rostro y se activara el botón de enviar, el cual hará el uso del GPS para indicar que se encontró una persona.</p>
<p>Evaluación de la Prueba:</p> <p>La prueba fue exitosa, las capturas se muestran en la figura A.2.</p>

En la figura A.2 se muestra la pantalla y el posicionamiento del botón Enviar en la aplicación al realizar el proceso de búsqueda.

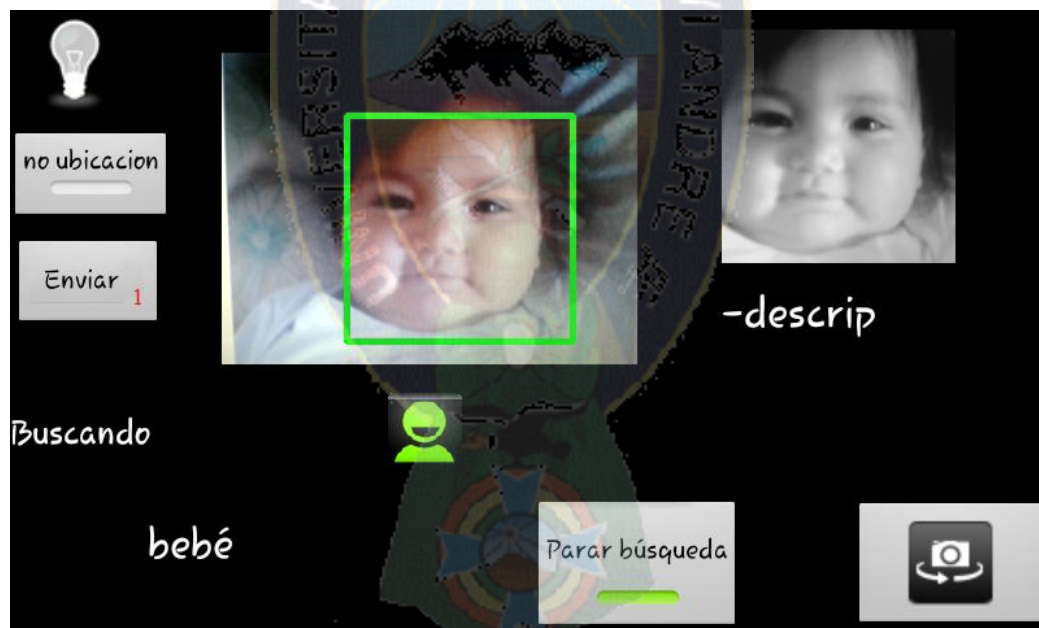


Figura A.2 Envío del mensaje de persona encontrada

1. Botón Enviar

Al realizar el proceso de búsqueda se procede al envío de datos cuando la persona escaneada coincide con una registrada, los datos enviados serán los siguientes:

- Nombre (coincidente al rostro escaneado)

- Descripción (coincidente al rostro escaneado)
- Latitud y longitud (obtenidos si el botón de ubicación se encuentra seleccionado)
- Precisión (Con respecto a latitud y longitud, rango 0 - 99)



Anexo F: Encuestas

Resultados de la encuesta ciudadana

En este anexo se muestran los resultados de la encuesta realizada a 30 personas a detalle, las preguntas pueden observarse en la tabla A.6.

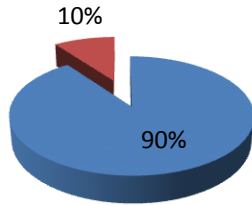
Tabla A.6 Resultados de la encuesta ciudadana

	Preguntas	Porcentaje opción (Mejora)	Porcentaje opción (No mejora, similar)
1	Estructura general del servicio móvil	27	3
2	Acceso a la aplicación móvil	4	26
3	Registro de nombre	7	23
4	Registro de datos de descripción	26	4
5	Registro de rostro de persona desaparecida	30	0
6	Registro de denuncia	28	2
7	Visualización de nombre al escaneo de rostro	14	16
8	Visualización de datos e información al escaneo de rostro	30	0
9	Opciones de localización	30	0
10	Método de identificación(escaneo)	27	3
11	Registro de Identificación	28	2
12	Tiempo de Registro	30	0
13	Tiempo de identificación	30	0
14	Intervalo de tiempo entre registro e Identificación(lapso de búsqueda)	28	2

Las siguientes graficas pertenecen a los resultados obtenidos

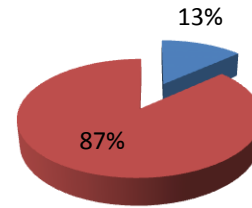
Pregunta 1: Estructura general del servicio móvil

- Porcentaje opción (Mejora)
- Porcentaje opción (No mejora, similar)



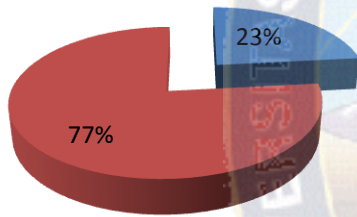
Pregunta 2: Acceso a la aplicación móvil

- Porcentaje opción (Mejora)
- Porcentaje opción (No mejora, similar)



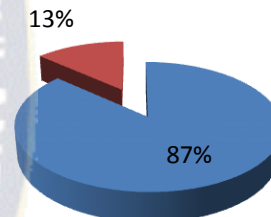
Pregunta 3: Registro de nombre

- Porcentaje opción (Mejora)
- Porcentaje opción (No mejora, similar)



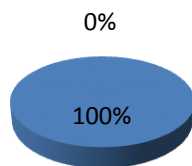
Pregunta 4: Registro de datos de descripción

- Porcentaje opción (Mejora)
- Porcentaje opción (No mejora, similar)



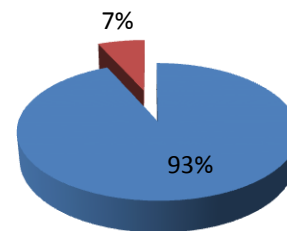
Pregunta 5: Registro de rostro de persona desaparecida

- Porcentaje opción (Mejora)



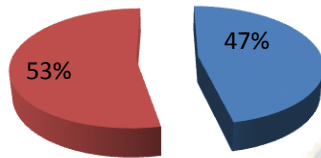
Pregunta 6: Registro de denuncia

- Porcentaje opción (Mejora)
- Porcentaje opción (No mejora, similar)



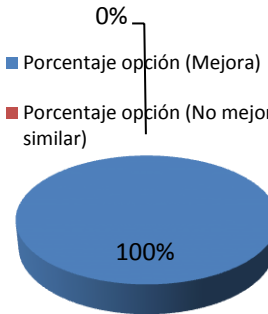
Pregunta 7: Visualización de nombre al escaneo de rostro

- Porcentaje opción (Mejora)
- Porcentaje opción (No mejora, similar)



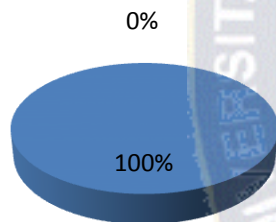
Pregunta 8: Visualización de datos e información al escaneo de rostro

- Porcentaje opción (Mejora)
- Porcentaje opción (No mejora, similar)



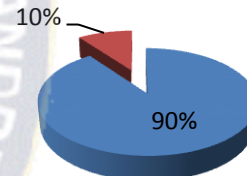
Pregunta 9: Opciones de localización

- Porcentaje opción (Mejora)
- Porcentaje opción (No mejora, similar)



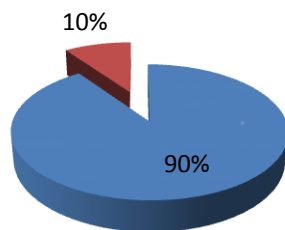
Pregunta 10: Método de identificación(escaneo)

- Porcentaje opción (Mejora)
- Porcentaje opción (No mejora, similar)



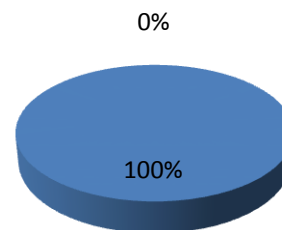
Pregunta 11: Registro de Identificación

- Porcentaje opción (Mejora)
- Porcentaje opción (No mejora, similar)



Pregunta 12: Tiempo de Registro

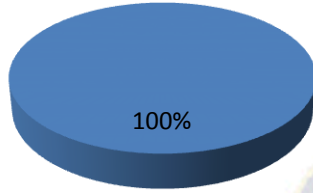
- Porcentaje opción (Mejora)
- Porcentaje opción (No mejora, similar)



Pregunta 13: Tiempo de identificación

- Porcentaje opción (Mejora)
- Porcentaje opción (No mejora, similar)

0%



Pregunta 14: Intervalo de tiempo entre registro e Identificación(lapso de búsqueda)

- Porcentaje opción (Mejora)
- Porcentaje opción (No mejora, similar)

7%

