

**Universidad Autónoma Metropolitana - Azcapotzalco**

**División de Ciencias Básicas e Ingeniería**

**Licenciatura en Ingeniería en computación**

**Aplicación móvil para el estudio anatómico del rostro por  
medio de realidad aumentada**

Proyecto tecnológico

Alan Garza Angeles

210340098

Asesor:

Leonardo Daniel Sánchez Martínez

Co-asesor:

José Alejandro Reyes Ortiz

Trimestre 2017 - Otoño

11 de diciembre de 2017

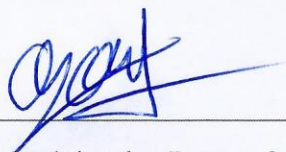
Yo, Leonardo Daniel Sánchez Martínez, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



---

Leonardo Daniel Sánchez Martínez

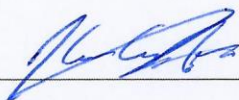
Yo, José Alejandro Reyes Ortiz, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



---

José Alejandro Reyes Ortiz

Yo, Alan Garza Angeles, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio de UAM Azcapotzalco.



---

Alan Garza Angeles

---

---

## Resumen

Este trabajo trata sobre la realización de una aplicación móvil para equipos con sistema operativo Android que, con apoyo de tecnología de realidad aumentada, permite al usuario interactuar y estudiar minuciosamente un modelo tridimensional del rostro humano junto con los diversos sistemas anatómicos que componen a este; esto con el objetivo de apoyar a los estudiantes de medicina y ciencias de la salud en el aprendizaje de la anatomía humana, siendo esta disciplina fundamental para la formación de profesionistas en esta área.

Para esto, se utilizó una metodología de desarrollo en cascada lo cual dio como resultado la implementación de los siguientes módulos, los cuales en conjunto permitieron el correcto funcionamiento de la aplicación: módulo de detección del rostro, módulo de extracción de información, módulo de transformación, módulo de realidad aumentada y módulo de interacción con el usuario.

La tecnología utilizada para la creación de la aplicación fue el motor gráfico de Unity y el kit de desarrollo Vuforia los cuales, por medio de la realización de scripts en el lenguaje de programación C#, permiten de forma sencilla la integración de elementos gráficos y lógicos para su posterior publicación en una amplia gama de plataformas.

En este trabajo se abordan los objetivos, los antecedentes y la justificación del proyecto realizado, los fundamentos teóricos y aspectos técnicos necesarios para el posterior diseño e implementación de la aplicación, así como los resultados y conclusiones obtenidos al finalizar la aplicación.

---

---

# Tabla de contenido

Introducción.....	1
Antecedentes.....	3
Justificación.....	5
Objetivos.....	6
Marco Teórico.....	7
1.1. Android.....	7
1.1.1. Historia.....	7
1.1.2. Características.....	7
1.1.3. Arquitectura.....	8
1.2. Realidad Aumentada.....	10
1.2.1. Definición.....	10
1.2.2. Historia.....	11
1.2.3. Elementos del sistema.....	14
1.2.4. Aplicaciones.....	14
1.3. Android y la Realidad Aumentada.....	16
1.4. Estudio de la Anatomía Humana.....	19
1.4.1. Ramas y divisiones.....	20
1.4.2. Sistemas y aparatos del cuerpo humano.....	20
1.4.3. Anatomía regional o topográfica.....	21
Desarrollo del proyecto.....	24
2.1. Componentes.....	24
2.1.1. Hardware.....	24
2.1.2. Software.....	25
2.2. Metodología.....	26
2.3. Desarrollo.....	26
Resultados.....	31
Análisis y discusión de resultados.....	35
Conclusiones.....	36
Referencias Bibliográficas.....	37
Apéndice A.....	40

---

---

## Índice de Figuras

1.1. Arquitectura del sistema Android . . . . .	8
1.2. El Sensorama de Morton Heilig . . . . .	11
1.3. Casco de realidad virtual inventado por Ivan Sutherland . . . . .	12
1.4. Louis Rosenberg en desarrollo de Virtual Fixtures . . . . .	12
1.5. KARMA, dispositivo de realidad aumentada para mantenimiento de impresoras . . . . .	13
1.6. Puntos elegidos por el Target Manager y nivel de aumentabilidad para realizar el reconocimiento de la imagen . . . . .	17
1.7. Sección de la cabeza del sistema óseo . . . . .	22
1.8. Sección de la cabeza del sistema muscular . . . . .	23
2.1. Funcionamiento del giroscopio dentro de un dispositivo móvil. . . . .	25
2.2. Diagrama de bloques de los módulos que conforman la aplicación móvil . . . . .	27
2.3. Diagrama que muestra la interacción del usuario con las vistas de la aplicación . . . . .	28
2.4. Modelos tridimensionales (de izquierda a derecha) del sistema óseo, el sistema muscular y el sistema tegumentario de un rostro, utilizados en la aplicación . . . . .	29
2.5. Diagrama de flujo que muestra el funcionamiento y la interacción entre los módulos de la aplicación . . . . .	30
3.1. Imagen seleccionada para el reconocimiento por parte de la aplicación (izquierda) y puntos característicos elegidos por Vuforia para dicho reconocimiento (derecha) . . . . .	31
3.2. Vista inicial del sistema antes del reconocimiento. . . . .	32
3.3. Vista en la que se ha hecho el reconocimiento de la imagen objetivo. . . . .	32
3.4. Diversas vistas con las que interactúa el usuario en donde se presentan los tres sistemas anatómicos disponibles . . . . .	33
3.5. Ejemplos de selecciones de elementos de los sistemas disponibles . . . . .	34

---

---

# Introducción

Una de las tecnologías de mayor auge en los últimos años es la de los dispositivos móviles, siendo que para noviembre del año 2016 se contabilizaban un total de 7,500 millones de teléfonos móviles con suscripción a un operador a lo largo del planeta, siendo de estos el 80% teléfonos inteligentes comparado al 70% del año anterior [1], demostrando un claro crecimiento en la demanda de aplicaciones para esta tecnología.

Esto indica que los teléfonos inteligentes o *smartphones* han revolucionado la manera en que interactuamos con la tecnología, siendo que estos no se adquieren solo por su capacidad de comunicación sino que se toman en cuenta otros elementos importantes como lo son: la cámara fotográfica y las especificaciones técnicas de la misma, la capacidad de almacenamiento para archivos y aplicaciones, el procesador y la RAM del dispositivo, así como diversos sensores que permiten a las aplicaciones interactuar con el entorno del dispositivo al conocer su posición en el espacio.

Estos elementos son los que permiten al dispositivo aprovechar al máximo la diversidad de tecnologías que se encuentran a disposición de los desarrolladores de apps<sup>1</sup> para dispositivos móviles, las cuales ofrecen un sinnúmero de servicios tales como comunicación, educación, compras, apoyo en tareas cotidianas, entretenimiento, entre otros.

Cabe mencionar, que este desarrollo tecnológico no es único de los teléfonos inteligentes, existiendo también otros tipos de dispositivos móviles, como las tabletas, que funcionan como el punto intermedio entre los *smartphones* y las computadoras personales, contando con la movilidad característica de los primeros, pero aportando también mejoras en cuanto a sus especificaciones técnicas y capacidades gráficas. La importancia de esto radica en que a medida que avanza la tecnología en el ámbito del hardware, la experiencia del usuario cambia con esta, surgiendo así nuevas formas de aprovechar sus beneficios por medio del desarrollo de software.

Una de las tecnologías aplicadas al desarrollo de apps con mayor auge en la actualidad es la realidad aumentada<sup>2</sup>, la cual, como su nombre lo señala, extiende la percepción del

---

<sup>1</sup> App o aplicación: Tipo de software que facilita al usuario la realización de tareas típicamente humanas, tales como gestionar una contabilidad o escribir un texto.

<sup>2</sup> La Realidad Aumentada (AR) consiste en sobreponer objetos o animaciones generadas por computadora sobre la imagen en tiempo real que recoge una cámara de un dispositivo. Es una tecnología que complementa la percepción e interacción con el mundo real y permite al usuario estar en un entorno aumentado con información generada por una computadora. [6]

usuario en cuanto a lo que puede observar con ayuda de la pantalla del dispositivo. Dicha tecnología irá en ascenso acorde al desarrollo de los dispositivos móviles, ya que esto permitirá mejorar la experiencia del usuario al lograr una mayor inmersión<sup>3</sup>, consecuencia de la mejora en procesamiento gráfico y la manera en la que ésta se presente al usuario, ejemplo de esto serían los visores y gafas de realidad aumentada como los HoloLens de Microsoft [2] y Glass de Google[3].

Las aplicaciones de la realidad aumentada se encuentran principalmente enfocadas al entretenimiento, sin embargo, la utilización de esta ha ido poco a poco abarcando un mayor número de áreas como lo son la publicidad, la mercadotecnia y la educación. Específicamente, se encuentra en la educación un gran potencial al poder utilizar los beneficios inmersivos que otorga la tecnología en función de apoyar a los estudiantes en procesos de memorización y aprendizaje.

Por otro lado, la anatomía<sup>4</sup> humana es de suma importancia en la formación de profesionistas en el área de la medicina ya que suministra al estudiante una serie de conocimientos, habilidades y actitudes que le serán esenciales para el futuro ejercicio de su profesión [4]. Sin embargo, esta disciplina representa un reto para los estudiantes que cursan dicha materia por lo extenso de su contenido, por lo que se requiere una adecuada técnica para el estudio y la comprensión de la misma. A su vez, se ha presentado un aumento en el uso de programas de simulación multimedia los cuales han favorecido la enseñanza y aprendizaje de la materia al ser ésta fundamentalmente visual [5].

En este trabajo se presenta el diseño e implementación de una aplicación móvil capaz de reconocer la imagen de un rostro preprocesada<sup>5</sup> en dos dimensiones y transformarla en un modelo tridimensional con el que el usuario puede interactuar por medio de la utilización de tecnologías de realidad aumentada; esto con el fin de contribuir en la mejora de la enseñanza y aprendizaje de la anatomía humana enfocada a estudiantes de medicina y ciencias afines a la salud, pedagogos y médicos jóvenes.

---

<sup>3</sup> Inmersión se puede entender como la acción de introducirse plenamente en un ambiente determinado, puede ser real o imaginario. [7]

<sup>4</sup> Ciencia que estudia la estructura y forma de los seres vivos y las relaciones entre las diversas partes que los constituyen. [7]

<sup>5</sup> Es decir, que dichas imágenes han sido digitalizadas y analizadas con anterioridad.

---

---

# Antecedentes

## Proyectos terminales

### 1. *Sistema de detección y reconocimiento facial multi-hilo sobre GPUs* [8]

En este proyecto se busca comparar el desempeño del algoritmo Local Binary Pattern, con el que se realizan reconocimientos faciales, cuando éste se ejecuta de forma secuencial o por medio de multi-hilos.

La tecnología utilizada es similar, ya que utiliza bibliotecas de OpenCV y trata temas de desempeño importantes para el desarrollo de mi proyecto. Sin embargo, los proyectos difieren en su producto final, siendo el mío una aplicación de uso específico y el de ellos un análisis de eficiencia de algoritmos.

### 2. *Cerradura electrónica con reconocimiento facial* [9]

La función de este proyecto es implementar un software que dote la capacidad a una cerradura electrónica el reconocimiento de rostros por medio de la utilización de una base de datos.

Del mismo modo, la tecnología es similar a la utilizada en mi proyecto, pero con una funcionalidad distinta. El objetivo de este proyecto es implementar la tecnología en función de controlar un dispositivo, mientras que mi proyecto se enfoca en implementarla en un sistema de realidad aumentada.

## Tesis

### 3. *Sistema de realidad aumentada para estrategias de mercadotecnia enfocada a una empresa inmobiliaria* [10]

Consiste en implementar un sistema en equipos Android haciendo uso de tecnologías de realidad aumentada con el objetivo de apoyar estrategias mercadotécnicas de una empresa mobiliaria, transformando una imagen arquitectónica de una casa en una estructura 3D digital.

La tecnología utilizada en el proyecto [10] es muy similar a la que utilizaré, en específico el software Unity y el kit de desarrollo Vuforia, siendo así una muy



buena referencia en cuanto a la implementación de tecnologías de realidad aumentada en un caso similar al de mi proyecto, sin la implementación del reconocimiento facial.

4. *Realidad aumentada aplicada a objetos de aprendizaje para asignaturas de Ingeniería Informática* [11]

La tesis [11] tiene como función implementar tecnología de realidad aumentada en objetos de aprendizaje, dirigido a las asignaturas de la carrera, proporcionando una alternativa interactiva de aprendizaje.

Mi proyecto comparte la justificación de la tesis, ya que ambas propuestas buscan mejorar los métodos de enseñanza, utilizando una tecnología similar, sin embargo, el área académica difiere, así como la implementación de la tecnología.

## **Software**

5. *Pharmagraphix* [12]

Es una aplicación móvil que por medio de reconocimiento de material bidimensional como lo son imágenes impresas y virtuales el dispositivo muestra un modelo tridimensional por medio de realidad aumentada. La aplicación cumple con varios de los objetivos del proyecto, con la diferencia de que mi proyecto busca no solo brindar un modelo tridimensional, sino además proveer información más detallada del modelo para el apoyo a estudiantes de medicina, permitiendo reconocer diferencias específicas de cada rostro y subsecuentemente mostrar el modelo tridimensional adecuado. Por otro lado, la aplicación [12] muestra el potencial que tiene el desarrollo de mi propuesta si se escala a un sistema anatómico más extenso.

6. *Anatomy 3D – Anatronica* [13]

Aplicación móvil que permite tener un nivel avanzado de interacción con un modelo anatómico en 3D por medio de una interfaz táctil. Comparten el mismo objetivo de mejorar el aprendizaje de la anatomía humana, y la tecnología a utilizar tiene ciertas similitudes, sobretodo en la interacción táctil con los modelos. Difieren en que la aplicación [13] no implementa realidad aumentada ni reconocimiento de elementos.

---

---

## Justificación

El desarrollo tecnológico ha fomentado una revolución en materia educativa, siendo que los estudiantes nacidos después de 1982 han tenido a lo largo de su vida contacto intenso con la tecnología [14]. Esta manera de aprendizaje por parte de los estudiantes de la generación actual se encuentra altamente enfocada a la utilización de tecnologías de información, comunicación y multimedia, y del mismo modo con el paso del tiempo, el sistema educativo se adapta a la realidad tecnológica dominante [15].

Esto muestra un claro crecimiento en la necesidad por herramientas tecnológicas que aprovechen el auge de esta plataforma informática móvil y que busquen mejorar la experiencia de los usuarios jóvenes que conviven con dicho auge tecnológico, con un enfoque en la educación.

Así, esta aplicación en conjunto con tecnologías multimedia como lo es la realidad aumentada permitirá complementar la percepción visual de los estudiantes de forma completamente computarizada, por un lado mostrando una solución atractiva al usuario y por otro ayudándolo en el proceso de aprendizaje de forma significativa al poder interactuar de forma inversiva con el modelo tridimensional y su entorno.

---

---

# Objetivos

## **Objetivo general:**

Desarrollar una aplicación móvil que permita generar modelos tridimensionales interactivos de rostros a partir de imágenes en dos dimensiones, con realidad aumentada.

## **Objetivos específicos:**

- Diseñar e implementar un módulo para la detección de un rostro a partir de una imagen válida.
- Diseñar e implementar un módulo para la extracción de información.
- Diseñar e implementar un módulo de transformación.
- Diseñar e implementar el módulo de realidad aumentada.
- Diseñar e implementar un módulo que permita la interacción con el modelo.

---

---

# Marco Teórico

En esta sección se abordará la descripción básica de los modelos teóricos de las tecnologías aplicadas en la realización de este proyecto.

## 1.1. Android

### 1.1.1. Historia

Android es un sistema operativo para dispositivos móviles basado en el núcleo Linux, desarrollado inicialmente por Android Inc., respaldado por Google y posteriormente comprado por éste. [16]. Android empezó su impulso gracias a la fundación de la *Open Handset Alliance*, un consorcio de hardware, software y operadores de servicio con el objetivo de desarrollar estándares abiertos para móviles y promover el diseño y difusión de la plataforma Android. Algunos de los integrantes de la fundación son Google, Intel, Texas Instruments, Motorola, T-Mobile, Samsung, Ericsson, Toshiba, Vodafone y otros [17].

Posteriormente al lanzamiento de la primera versión de Android SDK en el año de 2007, Android se consolida como uno de los sistemas operativos para móviles más utilizados en Estados Unidos, superando para el año de 2010 al sistema de Apple. En 2012, Google reemplaza Android Market por Google Play Store, donde en un solo portal se unifica la descarga de aplicaciones y contenidos de Android.

### 1.1.2. Características

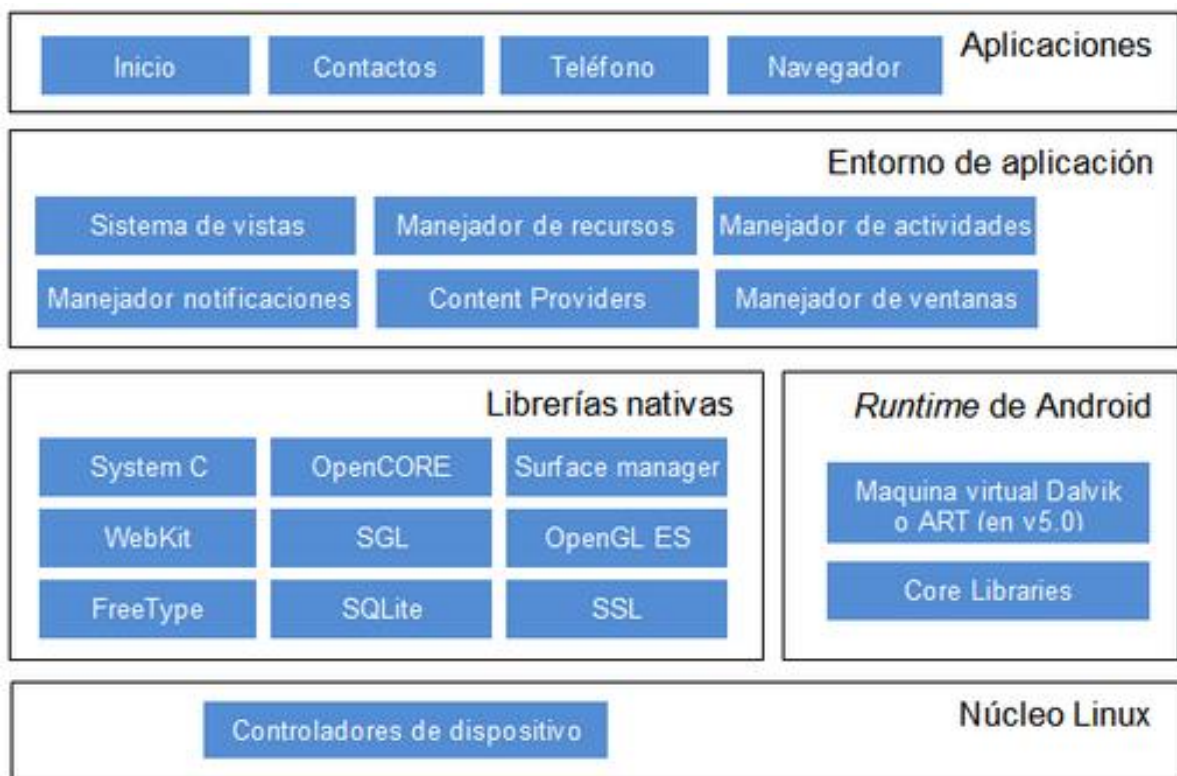
Entre las características que identifican a la plataforma Android se encuentran:

- Plataforma abierta. Es una plataforma de desarrollo libre basada en Linux y de código abierto.
- Portabilidad. Las aplicaciones finales son desarrolladas en Java, lo que asegura su ejecución en una amplia variedad de dispositivos.
- Arquitectura basada en componentes inspirados de Internet. La interfaz de usuario se hace por medio de código XML, permitiendo que se ejecute en distintos tipos de pantallas.
- Gran cantidad de servicios incorporados. Estos incluyen localización basada en GPS, bases de datos con SQL, reconocimiento y síntesis de voz, navegador, multimedia, etc.

- Seguridad. Cada aplicación dispone una serie de permisos que limitan su rango de actuación sobre el dispositivo.
- Optimizado para baja potencia y poca memoria. Android utiliza la Máquina Virtual Dalvik, la cual se trata de la implementación de Google de la máquina virtual de Java, optimizada para dispositivos móviles.
- Alta calidad de gráficos y sonido. Gráficos vectoriales suavizados, animaciones inspiraciones en Flash, gráficos en 3 dimensiones basados en OpenGL. Incorpora los códec estándares más comunes de audio y video como MP3, AAC, AVC, etc.

### 1.1.3. Arquitectura

La arquitectura de Android está formada por cinco capas, basadas en software libre, las cuales son: Aplicaciones, entorno de aplicación, bibliotecas, *runtime* de Android y núcleo Linux. Dichas capas se pueden observar en la Figura 1.1 y se describen a continuación [16].



**Figura 1.1.** Arquitectura del sistema Android.

#### Aplicaciones

En esta capa se encuentran todas las aplicaciones básicas con las que interactúa el usuario, como lo son clientes de correo electrónico, programas de SMS, calendario, mapas, navegador, contactos, etc. Todas las aplicaciones son ejecutadas en la máquina virtual Dalvik para garantizar la seguridad del sistema.

Normalmente, las aplicaciones Android son escritas en lenguaje Java utilizando la herramienta de desarrollo Android SDK, sin embargo, es posible también desarrollar las aplicaciones utilizando C/C++ con la utilización del Android NDK (*Native Development Kit*).

## Entorno de aplicación

Proporciona una plataforma de desarrollo libre para aplicaciones con gran variedad de tecnologías y componentes a disposición del desarrollador, como lo son sensores, localización, servicios, barra de notificaciones, etc. Así, esta capa permite la reutilización de componentes en el desarrollo de aplicaciones y aprovecha el lenguaje de programación, siendo compatible con una fracción muy significativa del entorno de ejecución Java.

Entre los servicios más importantes en el entorno de aplicación se encuentran:

- **Views:** Conjunto de vistas, es decir la parte visual de los componentes.
- **Resource Manager:** Proporciona acceso a recursos utilizados en el código.
- **Activity Manager:** Maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellos.
- **Notification Manager:** Permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
- **Content Providers:** Mecanismo que permite acceder a datos de otras aplicaciones, como los son contactos, archivos, etc.

## Librerías nativas

Incluye un conjunto de bibliotecas en C/C++ usadas por varios componentes del sistema. Se encuentran compiladas en el código nativo del procesador. Algunas de estas son:

- **System C library:** Derivación de la librería BSD de C estándar adaptada a dispositivos embebidos basados en Linux.
- **Media Framework:** Soporta codecs de reproducción y grabación de diversos formatos audio, video e imágenes como MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
- **Surface Manager:** Maneja el acceso al subsistema de representación gráfica en 2D y 3D.
- **WebKit:** Librería utilizada para navegación web, por navegadores como Google Chrome y Safari de Apple.
- **SGL:** Motor de gráficos 2D.

- **Librerías 3D:** Implementación basada en OpenGL. Las librerías utilizan el acelerador hardware 3D si está disponible, o el software altamente optimizado de proyección 3D.
- **FreeType:** Fuentes en bitmap y renderizado vectorial.
- **SQLite:** Potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.
- **SSL:** Proporciona servicios de encriptación *Secure Socket Layer*.

## Runtime de Android

Se basa en el concepto de máquina virtual utilizado en Java. Debido a las limitaciones existentes en memoria y procesador de los dispositivos móviles, no fue posible utilizar la máquina virtual de Java estándar, por lo que Google tomó la decisión de crear una nueva, la máquina virtual Dalvik, que respondiera mejor a estas limitaciones.

Esta máquina virtual permite la ejecución de ficheros Dalvik ejecutables (.dex), el cual es un formato optimizado para ahorro de memoria [18]. Además, está basada en registros, esto es que cada aplicación corre en su propio proceso de Linux con su propia instancia de la máquina virtual Dalvik, por lo que es posible correr múltiples máquinas virtuales de forma eficiente.

## Núcleo Linux

El núcleo del sistema está formado por el sistema operativo Linux. Esta capa proporciona servicios como la seguridad, gestión de memoria, gestión de multiprocesos, la pila de protocolos y el soporte de drivers para dispositivos. A su vez, actúa como una capa de abstracción entre el hardware y el resto de las capas de Android, por lo que es dependiente al hardware.

## 1.2. Realidad Aumentada

### 1.2.1. Definición

Realidad aumentada y realidad virtual son tecnologías relativamente recientes las cuales son usualmente confundidas entre ellas ya que buscan un objetivo común: maximizar la experiencia sensorial del usuario utilizando medios ajenos a la realidad. Esto es, por medio de diversas herramientas y tecnología, complementar la experiencia audiovisual del usuario creando una inmersión del mismo en un medio que no pertenece a la realidad. Sin embargo,

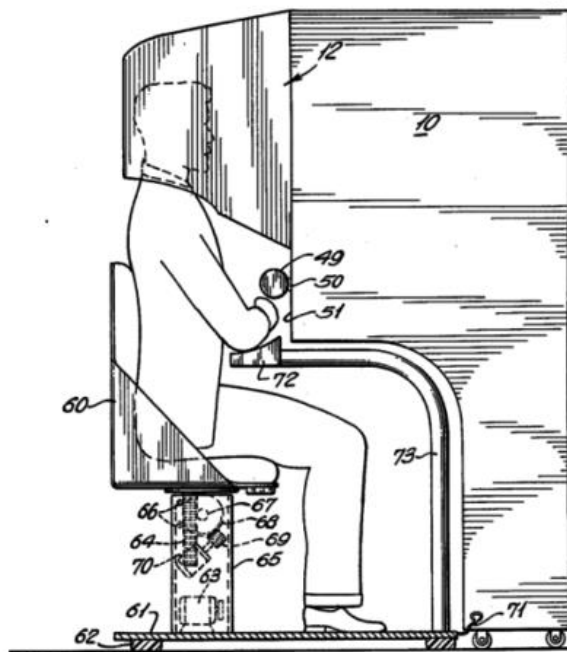
existen diferencias entre estos dos conceptos, las cuales han tomado mayor importancia con las tecnologías recientes.

Se puede definir a la realidad aumentada como la integración de información digital en el entorno del usuario en tiempo real, mientras que la realidad virtual crea por completo un entorno artificial, a diferencia de la realidad aumentada la cual usa entornos existentes y agrega nueva información sobre estos [19].

### 1.2.2. Historia

El término realidad aumentada es relativamente reciente, surgiendo en el año 1990 por el investigador Boeing Tom Caudell, implicado en el desarrollo de un software que permitía desplegar planos de cableado sobre las piezas producidas [20]. Sin embargo, el surgimiento de dicha tecnología es el resultado de una serie de investigaciones que se remontan a principios del siglo XX.

El primer invento que dio paso a lo que conocemos hoy en día como realidad aumentada, fue una máquina inventada por Morton Heilig, filósofo, inventor y cineasta, llamada Sensorama. Dicha máquina proyectaba imágenes en 3D, sumado a un sonido envolvente, un asiento que vibraba y creaba viento el cual podía percibir el espectador. Esto con el fin de simular la sensación de estar montado en una bicicleta, para esto se mostraba una grabación de la realidad en la que a medida que existía un cambio en el metraje la máquina aportaba los estímulos sensoriales correspondientes al usuario [21].



**Figura 1.2** El Sensorama de Morton Heilig [31]

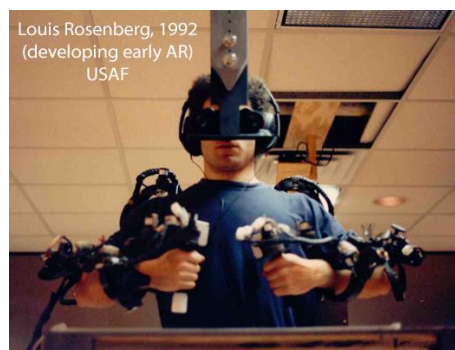


Si bien se considera a dicho inventor como el padre de la realidad virtual, ambas tecnologías se encuentran vinculadas desde su nacimiento y en ese entonces aún no se reconocía una diferencia entre ellas. En 1966 el profesor de Ingeniería Eléctrica de Harvard, Ivan Sutherland, crea el HMD (*Head Mounted Display*), el cual serviría como precursor de los visores utilizados actualmente para la realidad virtual y aumentada, aunque este dispositivo aún era una maquinaria de gran tamaño lo cual lo convertía en un dispositivo muy poco práctico y de difícil uso [22] como se puede observar en la Figura 1.3.



**Figura 1.3.** Casco de realidad virtual inventado por Ivan Sutherland [33]

Otro evento importante para el desarrollo de la realidad aumentada ocurre cuando se desarrolla lo que se considera como el primer sistema de realidad aumentada. L.B. Rosenberg, quien trabajaba para la fuerza área de Estado Unidos desarrolla en 1992 un dispositivo conocido como *Virtual Fixtures* el cual da consejos al usuario sobre cómo realizar ciertas tareas a medida que estas se presentan al piloto de forma virtual y a su vez publica un estudio acerca de cómo la realidad aumentada puede mejorar el desempeño humano [23].



**Figura 1.4.** Louis Rosenberg en desarrollo de Virtual Fixtures [34]

Por otro lado, en 1994 un equipo de científicos de la Universidad de Columbia liderado por Steven Feiner desarrollan un HMD que interactuaba con una impresora, bautizado como KARMA (*Knowledge-based Augmented Reality for Maintenance Assistance*), el cual proyecta una imagen en 3D que daba instrucciones al usuario acerca del mantenimiento y tareas de reparación de una impresora. [24]



**Figura 1.5.** KARMA, dispositivo de realidad aumentada para mantenimiento de impresoras [24]

Posterior al desarrollo de estos dispositivos se encuentra un claro crecimiento en la creación de nuevos dispositivos y software utilizado para esta tecnología, siendo ejemplos de esto el desarrollo de ARToolKit por Hirokazu Kato en 1999, un conjunto de librerías de código abierto para el desarrollo de aplicaciones de realidad aumentada en Windows, OS X, Linux, iOS y Android, el cual es aun ampliamente usado en la actualidad. En el 2012 sale al mercado Vuforia, el cual es un kit de desarrollo de software especializado en realidad aumentada, el cual permite desarrollar para plataformas iOS y Android, y a la vez cuenta con una extensión para el motor gráfico Unity<sup>6</sup>.

A su vez, el desarrollo de visores de realidad aumentada, herederos de los primeros HMD anteriormente mencionados, permite al usuario una aproximación más inmersiva de la realidad aumentada, al proporcionar una mayor comodidad y practicidad de su uso, lo cual potencializa los efectos sensoriales de la tecnología. Ejemplo de estos son los HoloLens de Microsoft y Glass de Google, lo cuales, si bien no han tenido aún un recibimiento comercial muy amplio, existen ya aplicaciones y propuestas innovadoras en desarrollo para dichas plataformas. En materia de software, una de las aplicaciones con mayor éxito ha sido Pokemon GO, un juego con elementos de realidad aumentada que alcanzó un éxito sin precedentes en el género, en parte por la premisa que dicha tecnología aportaba al juego [26].

---

<sup>6</sup> Motor gráfico multiplataforma utilizado para la creación de videojuegos, aplicaciones de realidad aumentada y realidad virtual. [25]

### 1.2.3. Elementos del sistema

Existen cuatro elementos de los que requiere un programa de realidad aumentada para funcionar, estos son: elemento capturador, elemento procesador, elemento de situación y el elemento sobre el cual proyectar [27].

- **Elemento capturador:**

Encargado de captar la imagen del mundo real o el elemento específico que funcione como entrada del programa de realidad aumentada. Usualmente, este elemento es una cámara digital que permite el reconocimiento de la imagen a procesar, pero puede aplicarse cualquier tipo de elemento que sea capaz de capturar imágenes, sonido o alguna señal.

- **Elementos de situación:**

Elemento que permite posicionar al objeto de salida virtual en el entorno real. Estos pueden ser por marcas o marcadores 2D y 3D en el entorno real, por sensores del dispositivo como el GPS, brújula, acelerómetro, giroscopio y por reconocimiento de objetos en bases de datos.

- **Elemento procesador:**

Parte del programa que permite interpretar la información recibida por el capturador y los elementos de situación acerca de los elementos reales, procesarla, crear los objetos virtuales que se desean agregar y combinarlos de la forma deseada. Para esto se requiere un módulo de reconocimientos de imágenes, de orientación espacial y superposición de imágenes.

- **Elemento sobre el cual proyectar**

Este elemento será el que permita mostrar el resultado de lo realizado por el procesador, es decir, la mezcla entre lo capturado del entorno real y el elemento virtual agregado. Este elemento se puede presentar como una pantalla de un dispositivo móvil, visores y HMD, así como por medio de proyecciones.

### 1.2.4. Aplicaciones

Para identificar a un programa de realidad como tal se busca que cumpla con ciertas características, como lo son que sea interactiva en tiempo real y que integre información

virtual dentro de un escenario real lo más realista posible. En base a esto se han realizado múltiples aplicaciones de estos programas en diversas disciplinas y ramas, como lo son:

- **Educación:**

Se han realizado aplicaciones de realidad aumentada para museos, exhibiciones y lugares históricos donde se presenta información de un objeto específico, así como elementos de interacción con estos. Por otro lado, también se han desarrollado aplicaciones específicas para el aprendizaje de materias, al poder presentar contenidos bajo premisas de entretenimiento y educación. Otra aplicación destacable son los traductores, que permiten sobreposicionar un texto de un determinado idioma sobre otro.

- **Televisión:**

Un claro ejemplo de aplicación televisiva es en deportes. La línea amarilla vista sobre el campo de juego en transmisiones de partidos de fútbol americano es un agregado visual que permite conocer el objetivo de la ofensiva para alcanzar el primero y diez, así como la agregación de otros elementos de análisis para este y otros deportes.

- **Entretenimiento:**

Una de las mayores aplicaciones de esta tecnología es en la industria de los videojuegos ya que esto permite nuevas posibilidades en la manera de jugar. Esto a su vez puede estar relacionado a alguna otra rama, como lo puede ser publicidad, mercadotecnia, educación, etc.

- **Arquitectura:**

La realidad aumentada ha permitido grandes innovaciones en el área de la arquitectura y de diseño de espacios, ya que esta permite visualizar el producto final permitiendo la integración de elementos de presentación de un proyecto, como los planos y maquetas. Por otro lado, permite resucitar virtualmente edificios históricos destruidos, así como proyectos de construcción.

- **Medicina:**

La realidad aumentada facilita el trabajo de doctores en campos como son la cirugía al permitir visualizar información adicional en momentos cruciales o al permitir practicar procedimientos con elementos virtuales. También es posible añadir información en elementos como radiografías y otras herramientas usadas para el diagnóstico de pacientes, existiendo incluso, unas gafas de realidad aumentada desarrollada por científicos de la Universidad de Washington, las cuales pueden distinguir las células cancerígenas de las sanas, marcando una revolución en los procedimientos quirúrgicos [28].

- **Publicidad:**

Otro campo importante de aplicación de la realidad aumentada es la publicidad y mercadotecnia, ya que por medio de las herramientas que proporciona esta tecnología es posible realizar campañas innovadoras y atractivas para al usuario. Uno de los elementos más usados para esto son los códigos QR, los cuales son variantes más potentes de los códigos de barras y que tiene como objetivo ser escaneados y ofrecer información extra de un producto específico.

### 1.3. Android y la Realidad Aumentada

La realidad aumentada ha encontrado en los dispositivos móviles una plataforma propicia para el aprovechamiento de ésta al contar con los elementos necesarios para su funcionamiento y a su vez con características favorables como la portabilidad y practicidad que estos dispositivos ofrecen. Aunado a esto, Android, siendo el sistema operativo líder en dichos dispositivos representa una plataforma de gran importancia para el desarrollo de aplicaciones de realidad aumentada.

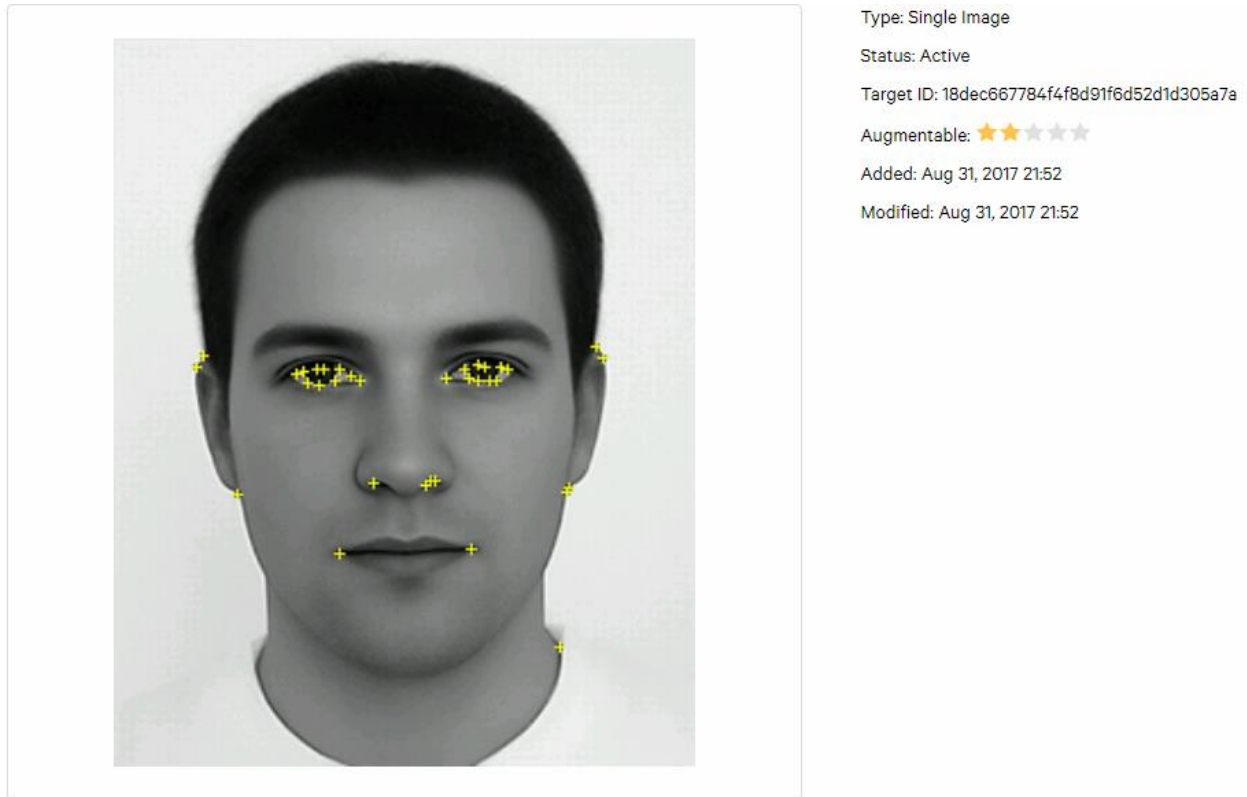
Vuforia es un kit de desarrollo de software de realidad aumentada para dispositivos móviles, siendo uno de los más usados a nivel mundial. Se utiliza tecnología de visión por computadora para realizar el reconocimiento de imágenes planas y objetos 3D simples como cajas y cilindros. Con ayuda de esto es posible desarrollar aplicaciones que posicionen y orienten objetos virtuales en relación al entorno del mundo real que es capturado por la cámara del dispositivo móvil. [29]

Los componentes principales de la herramienta de desarrollo son:

- **Motor Vuforia:** Librería del lado del cliente vinculado a la aplicación, permite desarrollar para Android, iOS y UWP por medio de la utilización de Android Studio, Xcode, Visual Studio o Unity.
- **Herramientas:** Vuforia proporciona una serie de herramientas a disposición del desarrollador, algunas de estas son:
  - *Vuforia Object Scanner:* ayuda escanear fácilmente objetos 3D en un formato compatible para el motor Vuforia.
  - *Target Manager:* Aplicación web en el portal de desarrollo que permite crear bases de datos de los objetivos a escanear en el dispositivo y en la nube. Como se muestra en la figura 1.6, al subir una imagen, la herramienta reconoce los puntos significativos de ésta los cuales permitirán realizar el reconocimiento,

añadiéndole un nivel de “aumentabilidad” que indica la facilidad con la que podrá realizar el reconocimiento de dicha imagen.

- *Calibration Assistant*: Permite a los usuarios finales crear perfiles personalizados que se adapten a la geometría del objetivo.



**Figura 1.6.** Puntos elegidos por el Target Manager y nivel de aumentabilidad para realizar el reconocimiento de la imagen.

Para el uso de las herramientas que ofrece Vuforia se requiere de la obtención de una llave de desarrollo proporcionada por el manejador de licencias de Vuforia.

- **Servicio de reconocimiento en la nube:** Vuforia ofrece este servicio para cuando la aplicación necesita reconocer una gran cantidad de imágenes o si la base de datos es frecuentemente actualizada.

Unity es un motor gráfico para creación de videojuegos multiplataforma que permite agregar las herramientas de Vuforia para el desarrollo de aplicaciones de realidad aumentada de forma simple. Unity es notable por su capacidad multiplataforma, lo cual lo vuelve una gran opción a la hora del desarrollo de aplicaciones y videojuegos, siendo especialmente usado para el desarrollo de aplicaciones en dispositivos móviles. Las plataformas soportadas por Unity son las siguientes:

- Web
  - Facebook Gameroom

- PC
  - Windows
  - Universal Windows Platform
  - SteamOS
  - OS X
  - Linux
- Dispositivos Móviles
  - iOS
  - Android
  - Windows Phones
  - Tizen
  - Fire OS
- Smart TV
  - tvOS
  - Samsung Smart TV
  - Android TV
- Consolas
  - PlayStation Vita
  - PlayStation 4
  - Xbox 360
  - Xbox One
  - Wii U
  - Nintendo Switch
  - Nintendo 3DS
- Dispositivos de Realidad Aumentada y Virtual
  - Oculus Rift
  - Google Cardboard
  - HTC Vive
  - PlayStation VR
  - Samsung Gear VR

### **Ventajas de Unity/Vuforia:**

- Las aplicaciones de Unity y Vuforia se encuentran en constante actualización.
- El manejo de objetos 3D se simplifican enormemente al contar con una interfaz gráfica para el manejo de estos.
- Unity cuenta con módulos que facilitan el desarrollo.
- Unity tiene modelos ya prediseñados.

- Existe una gran cantidad de *assets* a disposición en línea, que permite la obtención de modelos y demás componentes tridimensionales gratis o con costo.
- Desarrollar en Unity permite desarrollar para múltiples plataformas simultáneamente.
- Unity al ser un motor gráfico, agrupa diversas etapas de desarrollo en un solo entorno.
- Gran sinergia entre el código y el diseño gráfico de la aplicación.
- Vuforia permite de una manera sencilla crear aplicaciones de realidad aumentada ya que cuenta con scripts precargados para la realización de las acciones básicas.

#### **Desventajas:**

- Se tiene poco margen de maniobra al querer implementar nuevas funciones y acciones de realidad aumentada sin conocimiento de los scripts de Vuforia.
- Solo se pueden crear scripts en JavaScript y en C#.
- Unity y Vuforia cuentan con poca documentación en español.
- Las aplicaciones creadas con Unity tienden a ser más pesadas que al crearlas con un entorno de desarrollo nativo.
- Se requiere mayor capacidad gráfica por parte del dispositivo.
- Unity y Vuforia son de cobro si el producto final se comercializa.

## **1.4. Estudio de la Anatomía Humana**

La anatomía humana es un campo especial de la anatomía general enfocada al estudio de la estructura y forma de los seres humanos y las relaciones entre las diversas partes que los constituyen [7]. Bajo una visión sistemática, el cuerpo humano está organizado en diferentes niveles según una jerarquía: El cuerpo se compone de aparatos, éstos se integran de sistemas, que a su vez están compuestos por órganos, los cuales se componen de tejidos, estos se integran por células, que a su vez se forman de moléculas.

La anatomía constituye la base de la práctica de la medicina. Esta lleva al médico a la comprensión de la enfermedad del paciente al realizar una exploración física a través de ésta. También es de gran utilidad para dentistas, quiropráctico, fisioterapeutas y todos aquellos implicados en cualquier forma de tratamiento de pacientes cuyo primer paso sea el análisis de signos clínicos, así, la capacidad de interpretar una observación clínica es el punto final de una comprensión anatómica profunda [30].



### 1.4.1. Ramas y divisiones

Existen varias formas de aproximarse al estudio de la anatomía humana, algunas de estas son:

- **Anatomía sistemática o descriptiva:** Se estudia y se sigue cada sistema a través de todo el cuerpo. Fracciona al cuerpo humano organizándolo por sistemas y aparatos.
- **Anatomía regional o topográfica:** Se estudia cada región del cuerpo por separado y todos los aspectos de dicha región se estudian al mismo tiempo.
- **Anatomía artística:** Trata de las cuestiones anatómicas que afectan directamente a la representación artística de la figura humana.
- **Anatomía clínica:** Es un estudio anatómico enfatizado a situaciones de índole médico-clínica, así como otras ciencias de la salud.

### 1.4.2. Sistemas y aparatos del cuerpo humano

En el estudio de la anatomía, un sistema es un conjunto de órganos relacionado entre ellos que convergen en una función general y que comúnmente están formados predominantemente por los mismos tipos de tejido, como lo es el sistema muscular, a diferencia de los aparatos, los cuales no tienen un tipo de tejido predominante en los órganos que lo conforman, como lo son el aparato locomotor el cual se conforma por los sistemas muscular, óseo y articular.

- **Sistema inmunitario:** Provee defensas contra agentes causantes de enfermedades.
- **Sistema tegumentario:** Se conforma de la piel, vellos y uñas.
- **Sistema nervioso:** Formado por el sistema nervioso central (encéfalo<sup>7</sup> y médula espinal).
- **Sistema circulatorio:** Formado por el corazón, arterias, venas y capilares.
- **Sistema linfático:** Formado por los capilares, caos y ganglios linfáticos, bazo, timo y médula ósea.
- **Sistema endócrino:** Está formado por las glándulas endócrinas y órganos que segregan hormonas.
- **Sistema muscular:** Formado por los diversos músculos del cuerpo que permiten el movimiento del mismo.

---

<sup>7</sup> Parte central del Sistema nervioso encerrada y protegida en la cavidad craneal, formado por el cerebro, el cerebelo y el bulbo raquídeo.

- **Sistema óseo:** Conformado por los diversos huesos en el cuerpo, proporcionan apoyo estructural y protección a los órganos.
- **Sistema articular:** Formado por las articulaciones y ligamentos asociados que unen el sistema óseo y permite los movimientos corporales.
- **Aparato digestivo:** Se compone por la boca, faringe, esófago, estómago, intestinos y glándulas anexas. Permite la conversión de alimento en moléculas asimilables mediante enzimas.
- **Aparato urinario:** Eliminación de sustancias tóxicas y desechos del cuerpo mediante la orina.
- **Aparato reproductor:** Se compone por los órganos sexuales masculinos o femeninos.
- **Aparato respiratorio:** Órganos y cavidades empleadas para el intercambio de gases respiratorios.
- **Aparato locomotor:** Conjunto de los sistemas óseo, articular y muscular. Este sistema, coordinado por el sistema nervioso, permite la locomoción.
- **Aparato cardiovascular:** Conjunto de vasos sanguíneos y linfáticos.

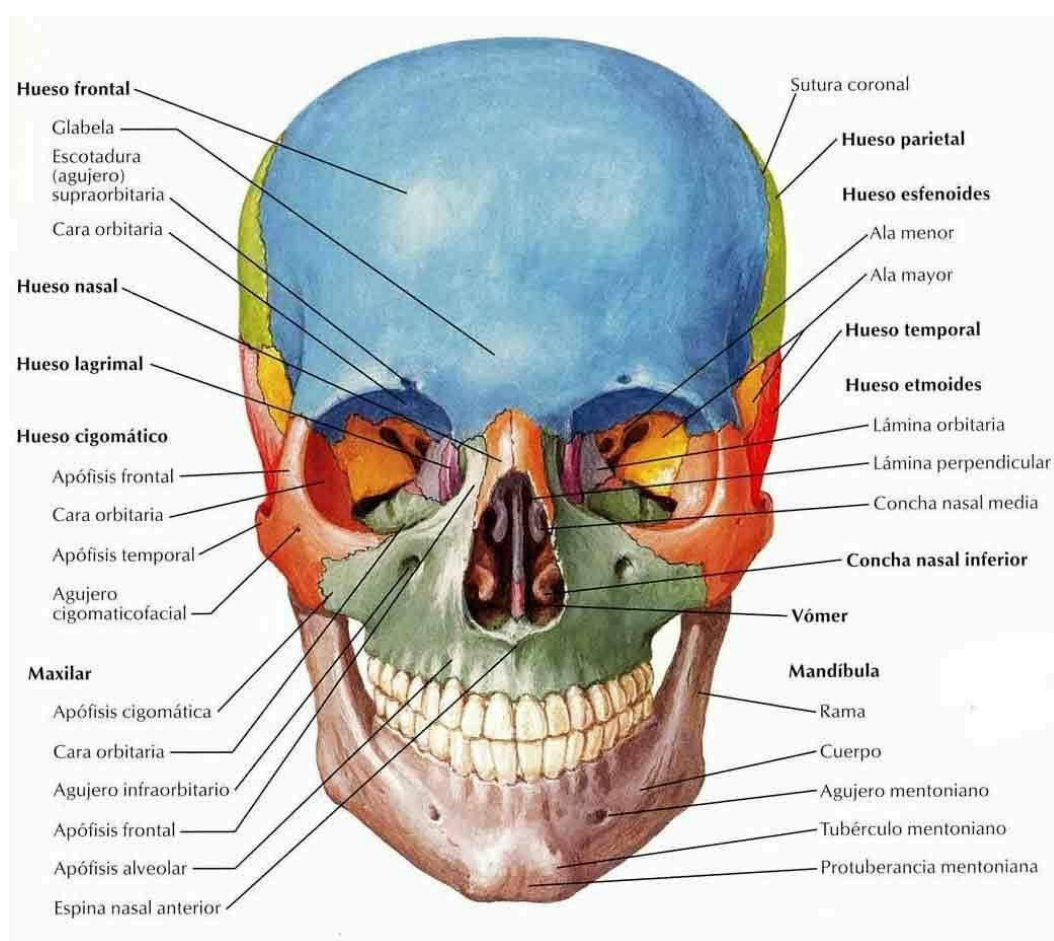
#### 1.4.3. Anatomía regional o topográfica

Bajo el estudio de la anatomía humana en un criterio topográfico, se divide el cuerpo humano en regiones esquemáticas, estas son:

- Cabeza (*Testa*)
  - Neurocráneo (*Neurocranium*)
  - Viscerocráneo (*Viscerocranium*)
- Cuello (*Collum*)
- Tronco (*Truncus*)
  - Espalda (*Dorsum*)
  - Tórax (*Thorax*)
  - Abdomen (*Abdomen*)
  - Pelvis (*Pelvis*)
- Miembro superior (*Membrum superius*)
  - Cintura escapular (*Cingulum membri superioris*)

- Brazo (*Brachium*)
- Antebrazo (*Antebrachium*)
- Mano (*Manus*)
- Miembro inferior (*Membrum inferius*)
  - Cintura pelviana (*Cingulum membri inferioris*)
  - Muslo (*Fémur*)
  - Pierna (*Crus*)
  - Pie (*Pes*)

De este modo, es posible seccionar elementos de los diversos sistemas y aparatos que conforman al cuerpo por medio del uso de la anatomía topográfica, simplificando y esquematizando el estudio del mismo. Ejemplo es la jerarquización de los huesos y músculos que conforman el área de la cabeza como se muestra en la Figura 1.7 y 1.8.



**Figura 1.7.** Sección de la cabeza del sistema óseo [35]



**Figura 1.8.** Sección de la cabeza del sistema muscular [33]

---

---

# Desarrollo del proyecto

## 2.1. Componentes

Para el desarrollo del proyecto se requirieron de elementos de software y hardware específicos los cuales permitieron la realización del mismo. A continuación se describen dichos componentes.

### 2.1.1. Hardware

En cuanto a hardware, se requirió únicamente de una computadora capaz de soportar el programa de Unity 5 y Visual Studio 2015, dichos requerimientos son los siguientes:

**OS:** Windows 7 SP1+, 8, 10, 64-bit versions only; Mac OS X 10.9+.

Windows XP y Vista no son compatibles.

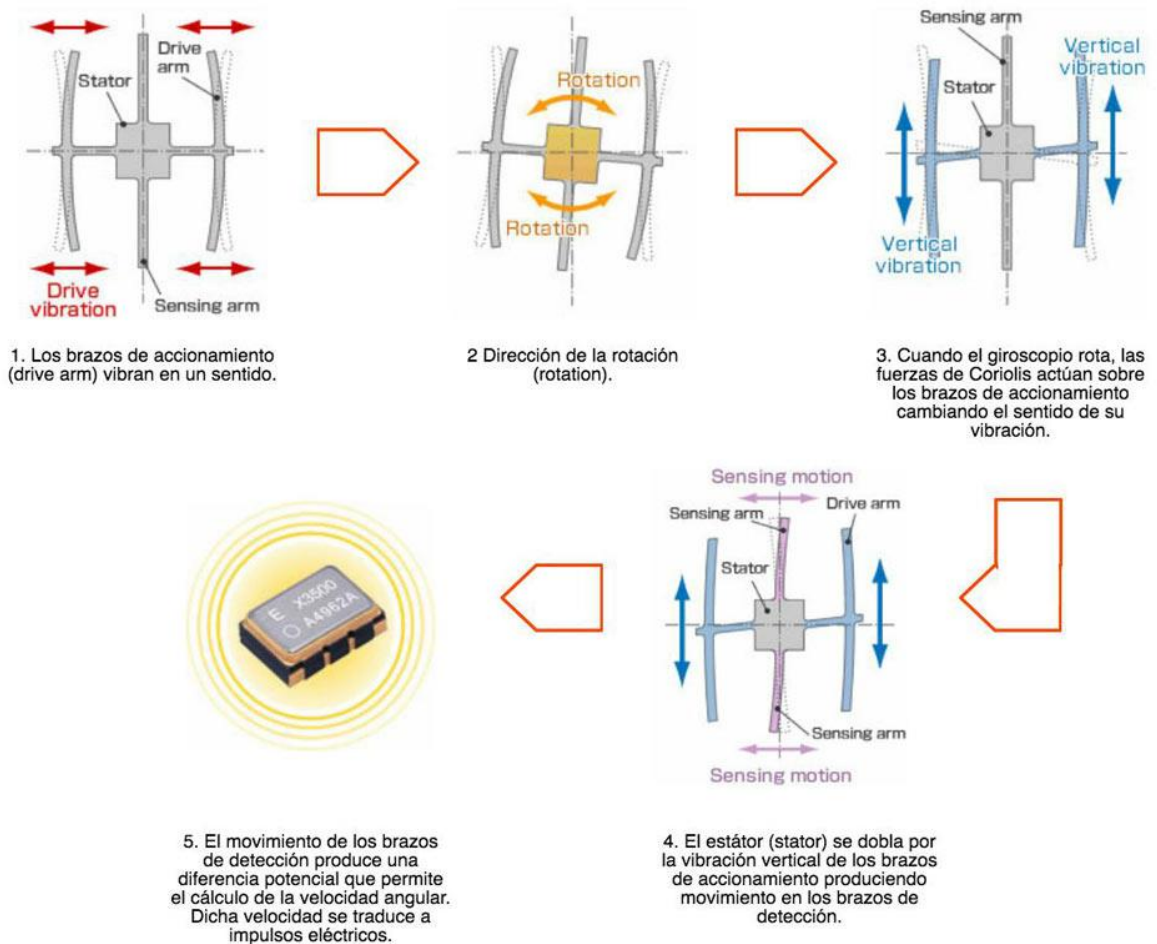
**CPU:** Soporte para el conjunto de instrucciones SSE2.

**GPU:** Tarjeta gráfica con DX9 (modelo de shader 3.0) o DX11 con capacidades de funciones de nivel 9.3.

Requisitos adicionales para el desarrollo de plataformas:

- iOS: Computadora Mac con sistema operativo versión OS X 10.9.4 como mínimo y Xcode 7.0 o superior.
- Android: Android SDK y Java Development Kit (JDK); IL2CPP scripting backend requiere Android NDK.
- Windows Store: Windows 8.1 (64 bits) y Visual Studio y plataforma SDK correspondiente:
  - Universal Windows Platform (UWP): Windows 8.1 (64 bits), Visual Studio 2015 o posterior y Windows 10 SDK;
  - El scripting backend IL2CPP también requiere la instalación de la función de compilador C++ con Visual Studio.

En cuanto al dispositivo móvil, se requiere que este cuente con giroscopio en orden de poder ejecutar aplicaciones de realidad aumentada. Esto debido a que gracias a este sensor es posible para el *smartphone* saber en qué plano del espacio se encuentra, cuál es su aceleración con respecto al punto de partida y si el dispositivo está girando sobre el plano o inclinándose, es decir, detecta el movimiento en sus seis ejes. Esto permite que en tecnologías como la realidad virtual y aumentada sea posible cuadrar la imagen en pantalla con la posición y movimiento del usuario. El funcionamiento del giroscopio se puede observar en la Figura 2.1.



**Figura 2.1.** Funcionamiento del giroscopio dentro de un dispositivo móvil [36]

### 2.1.2. Software

Para el desarrollo de la aplicación se hizo uso del motor gráfico Unity versión 5.6.2 junto con el kit de desarrollo Vuforia versión 6.2.10 para Unity. En cuanto al desarrollo de modelos tridimensionales se hizo uso del software Autodesk Maya 2018. A su vez, para el

desarrollo de los scripts en lenguaje C# que permiten la interacción con los modelos tridimensionales, se usó el entorno de desarrollo integrado Visual Studio 2015.

## 2.2. Metodología

La metodología aplicada para el desarrollo del proyecto fue la conocida como método de cascada, llamada así por ser un enfoque sistemático y secuencial y por lo tanto, para comenzar con una etapa se requiere la finalización de la etapa anterior [31]. Este método consta de cinco etapas las cuales se describen a continuación:

- 1. Comunicación:** Inicio del proyecto, se lleva a cabo el análisis de requerimientos del usuario. De esta etapa se obtienen los objetivos que debe cubrir el proyecto.
- 2. Planeación:** Etapa de estimación, programación y seguimientos de los diversos módulos del sistema. Contiene la descripción de la estructura global del sistema y la especificación de lo que debe hacer cada una de sus partes.
- 3. Modelado:** Etapa de análisis y diseño del sistema. Se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario, así como los análisis necesarios para definir las herramientas a usar en la etapa de construcción.
- 4. Construcción:** Se implementa el código fuente por medio de las herramientas definidas en la etapa anterior, así como la realización de pruebas y ensayos necesarios para la corrección de errores.
- 5. Despliegue:** Etapa final del desarrollo donde se realiza la entrega del proyecto y se recibe retroalimentación por parte del usuario final.

## 2.3. Desarrollo

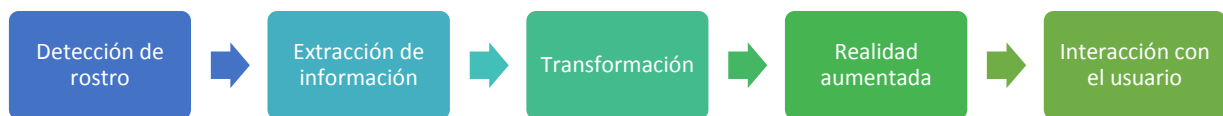
En función del punto anterior, a continuación, se describe el proceso realizado para el desarrollo del sistema:

Inicialmente, se establecieron los requerimientos que tendría que cumplir el sistema para considerarse terminado, estos son:

- La aplicación será capaz de identificar una imagen preprocesada y transformarla en un modelo tridimensional de la misma.

- Posteriormente, el usuario será capaz de seleccionar diversos sistemas anatómicos del modelo tridimensional, así como aislar elementos de cada sistema y conocer información adicional acerca del elemento seleccionado.
- Con ayuda de la realidad aumentada el usuario podrá redimensionar el modelo y observar distintas perspectivas de éste al momento de mover el dispositivo hacia la dirección deseada por el usuario.
- La interfaz gráfica debe ser lo suficientemente intuitiva para que el usuario pueda realizar dichas acciones de forma sencilla.
- La aplicación deberá funcionar de manera fluida y encontrarse optimizada para su ejecución en dispositivos móviles.

Una vez establecidos los requerimientos funcionales y no funcionales, es posible establecer los módulos que conformarán al sistema, los cuales se pueden observar en la Figura 2.2 y se describen a continuación.



**Figura 2.2.** Diagrama de bloques de los módulos que conforman la aplicación móvil.

- **Módulo de detección de rostro**

Este módulo es el primer componente interno del sistema y se conforma por los elementos necesarios para poder detectar una imagen válida para el sistema. Por medio de este módulo, el usuario será capaz de detectar si la imagen introducida es una imagen válida para el sistema o no. Este módulo debe detectar un rostro a partir de una imagen preprocesada.

- **Módulo de extracción de información**

Este módulo será el responsable de recibir la información obtenida por el módulo de detección de rostro. Se encargará de recuperar el número de bordes y relieves de una imagen preprocesada en dos dimensiones, las cuales definirán las dimensiones y características específicas de un rostro en particular.

- **Módulo de transformación**

Este módulo será el encargado de convertir un plano de bordes y relieves a una imagen tridimensional. Para ello, tomará como entrada los datos entregados por el módulo de



extracción de información y extrapolará<sup>8</sup> estos para determinar las características del modelo tridimensional.

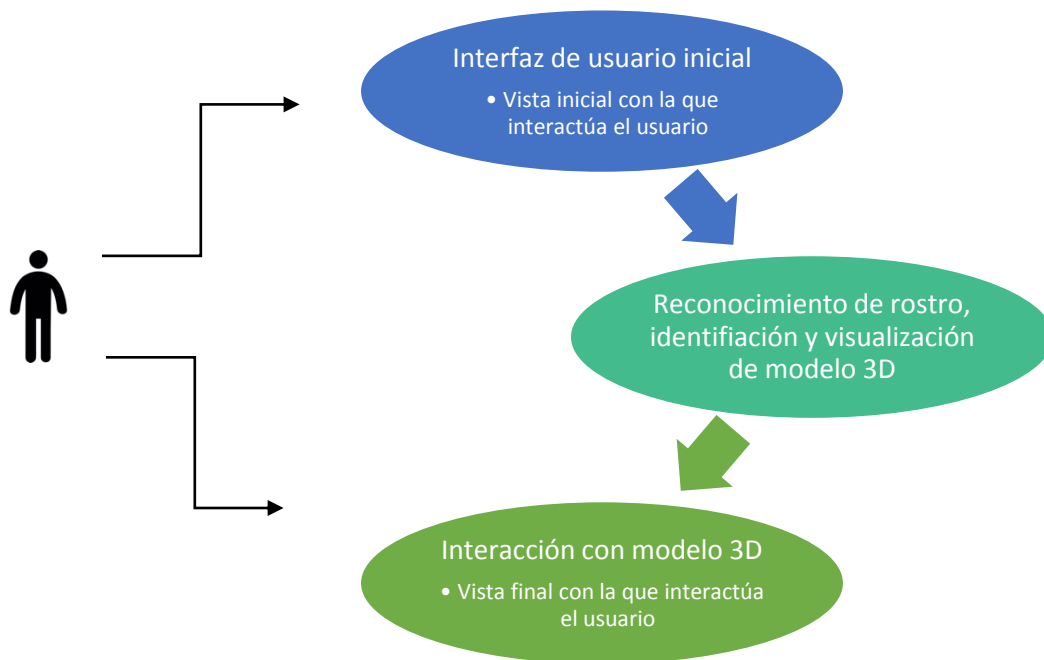
- **Módulo de realidad aumentada**

Este módulo permitirá agregar información adicional al modelo tridimensional generado por el módulo de transformación. La información se agregará de manera superpuesta de modo que el modelo contendrá información detallada de cada parte del mismo.

- **Módulo de interacción con el usuario**

El objetivo de este módulo es permitir al usuario interactuar con el modelo tridimensional durante tiempo de ejecución. Esto quiere decir, que una vez que el objeto tridimensional se visualice en pantalla, el usuario será capaz de realizar ciertas acciones sobre los elementos que conformen al modelo.

La Figura 2.3 muestra la manera en la que el usuario interactúa con los módulos planteados para la aplicación, mientras que en la Figura 2.5 se visualiza de forma detallada el algoritmo que la aplicación sigue para el funcionamiento de la misma.

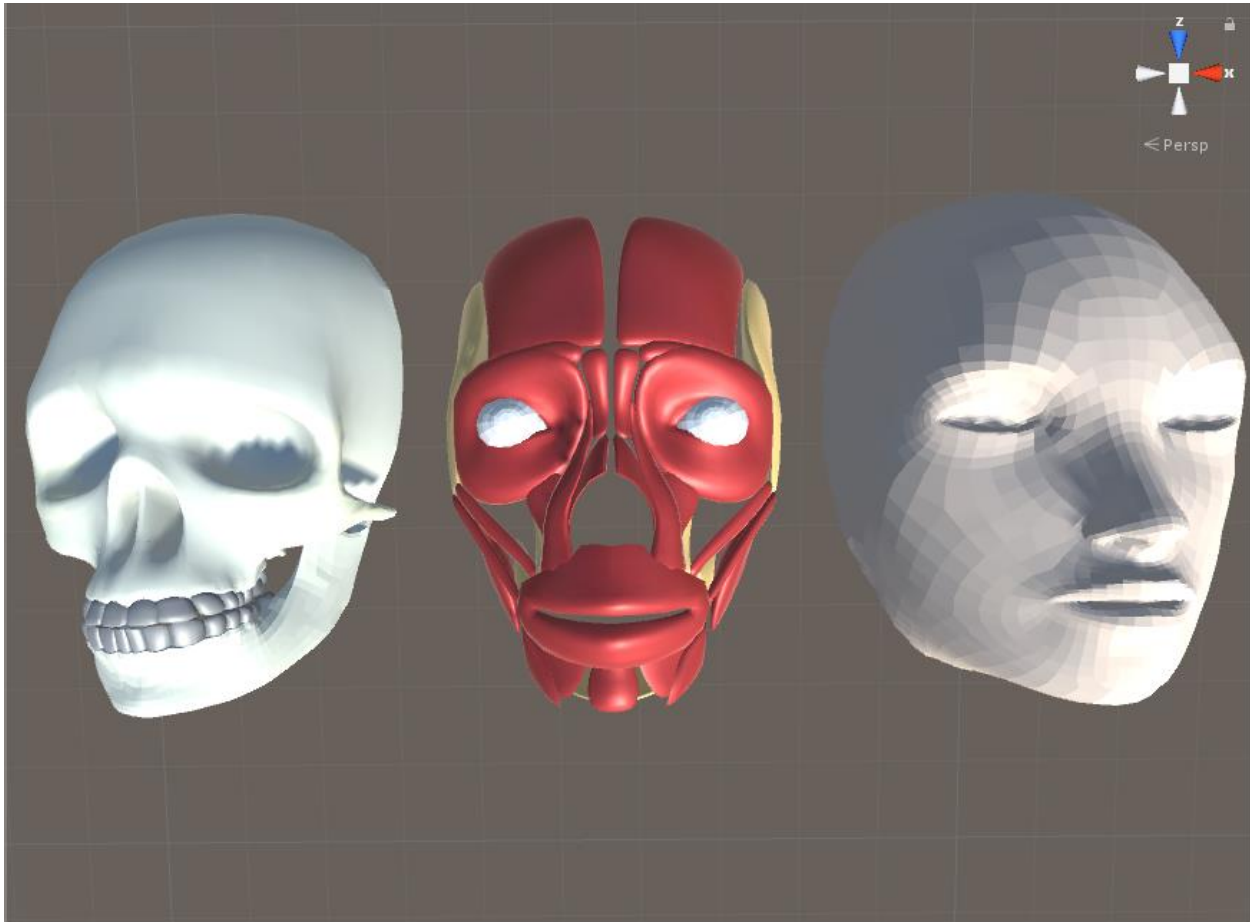


**Figura 2.3.** Diagrama que muestra la interacción del usuario con las vistas de la aplicación

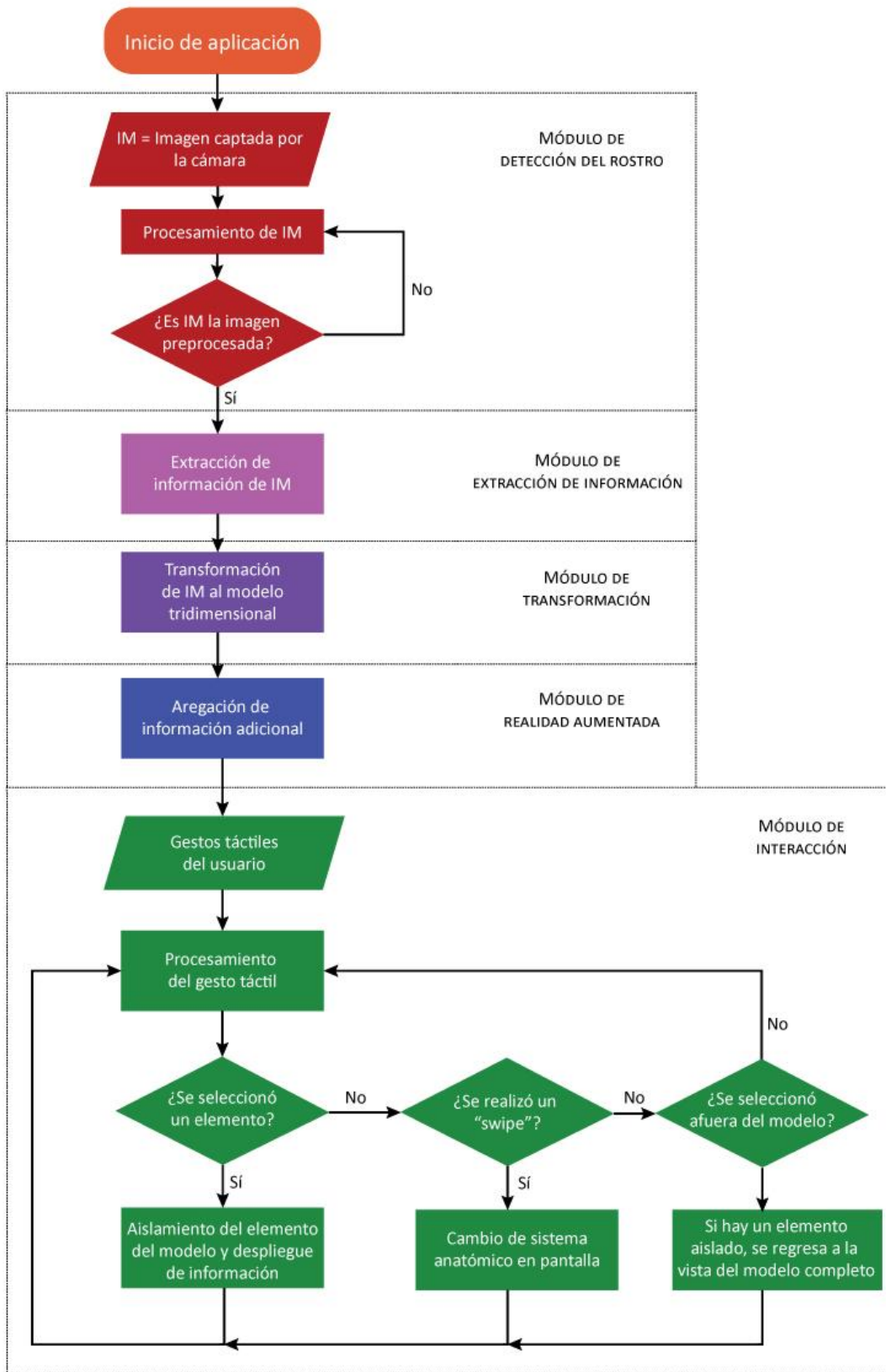
---

<sup>8</sup> Aplicar a un ámbito determinado conclusiones obtenidas en otro.

Así, una vez definidos los módulos y establecidas las herramientas a utilizar (presentadas en la subsección 2.1 del presente capítulo), es posible continuar a la construcción del sistema por medio del desarrollo de los modelos tridimensionales y la codificación de los módulos. El resultado de los modelos tridimensionales usados en la aplicación se puede observar en la Figura 2.4, mientras que el código fuente de la aplicación, el cual está compuesto de scripts escritos en lenguaje C#, se encuentran enlistados en el Apéndice A de este documento.



**Figura 2.4.** Modelos tridimensionales (de izquierda a derecha) del sistema óseo, el sistema muscular y el sistema tegumentario de un rostro, utilizados en la aplicación.



**Figura 2.5.** Diagrama de flujo que muestra el funcionamiento y la interacción entre los módulos de la aplicación

---

---

## Resultados

Una vez integrados los módulos establecidos, la aplicación es capaz de identificar una imagen almacenada en la base de datos que proporciona Vuforia y realizar el agregado de los elementos de realidad aumentada sobre esta. La imagen elegida y los puntos característicos para el reconocimiento de ésta se pueden apreciar en la Figura 3.1. Se realizó la elección de esta imagen para las pruebas por su simetría, lo cual facilitó colocar los elementos tridimensionales sobre el rostro permitiendo cierta continuidad y naturalidad entre los elementos.



**Figura 3.1.** Imagen seleccionada para el reconocimiento por parte de la aplicación (izquierda) y puntos característicos elegidos por Vuforia para dicho reconocimiento (derecha)

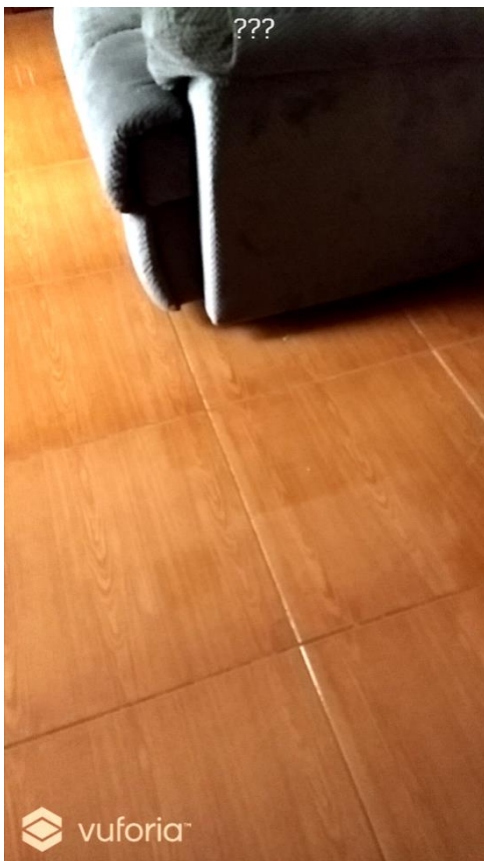
A pesar de que el nivel de aumentabilidad que nos proporciona la base de datos acerca de la imagen es muy bajo (de sólo una estrella) el reconocimiento por parte de la aplicación se realiza de forma rápida en la mayoría de los casos. A continuación se presentan las pantallas obtenidas de la ejecución del programa.

La primera vista con la que el usuario se encuentra al inicio de la aplicación es la que sucede antes del reconocimiento de la imagen objetivo, en esta vista únicamente se observa lo que la cámara se encuentre enfocando en el momento junto con tres signos de

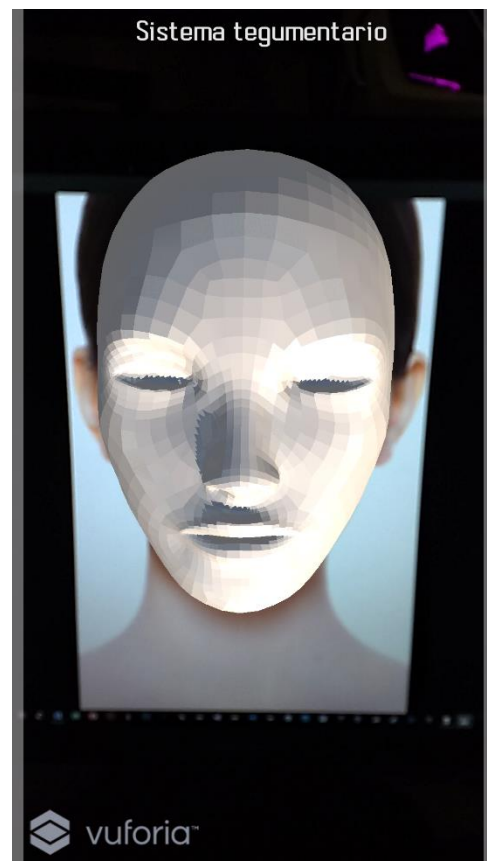
interrogación en la parte de arriba de la vista que indican que no se ha reconocido ningún objeto. Esto se puede observar la Figura 3.2.

Al momento que la cámara enfoca la imagen objetivo, se avanza a la siguiente vista de la aplicación donde ya se encuentra los elementos de realidad aumentada agregados sobre la imagen objetivo captada por el dispositivo, como se muestra en la Figura 3.3. Estos serán visibles siempre que se mantenga la cámara enfocada sobre la imagen, de lo contrario se regresará a la vista de la Figura 3.2.

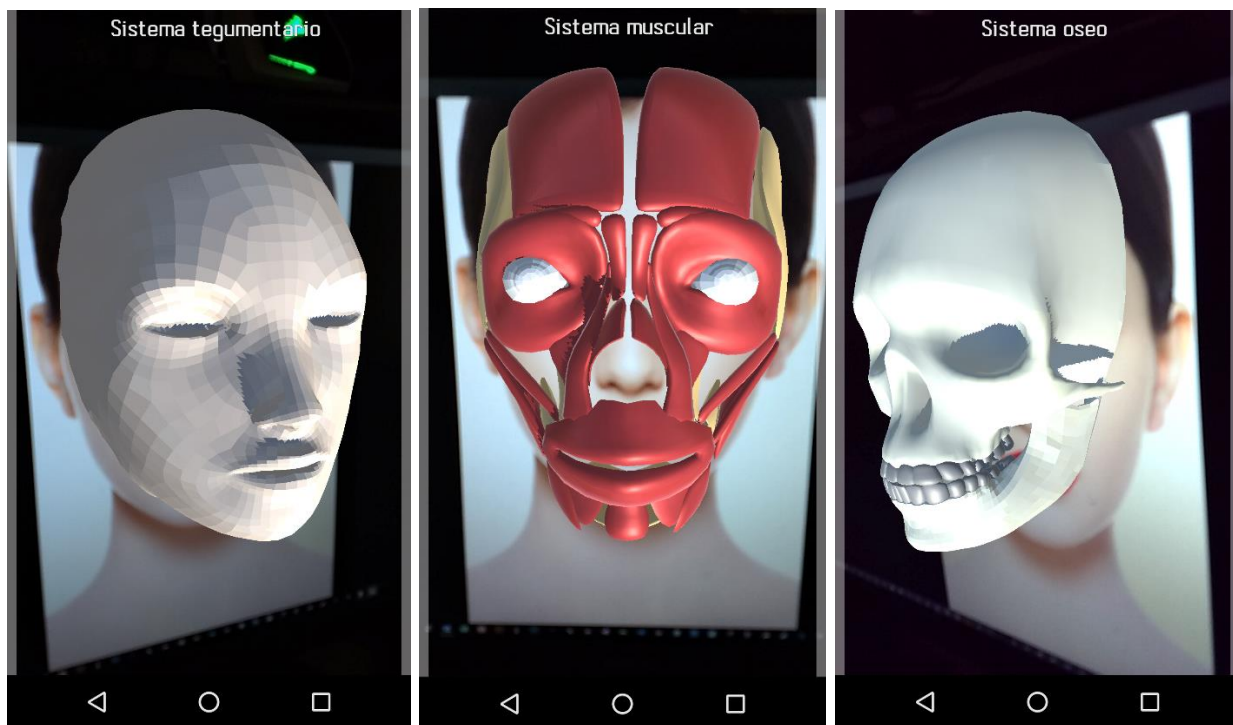
Durante esta vista, el usuario es capaz de interactuar con los modelos tridimensionales presentados. Por medio de *swipes* sobre la pantalla es posible intercambiar entre los tres sistemas disponibles en la aplicación mostrando el nombre del sistema en la parte superior de la pantalla, como se muestra en la Figura 3.4, a su vez, si se selecciona algún elemento de estos por medio de un *touch* sobre la pantalla, el elemento se aislará y mostrará el nombre de dicho elemento en la parte inferior, ejemplo de esto se muestra en la Figura 3.5.



**Figura 3.2.** Vista inicial del sistema, antes del reconocimiento.



**Figura 3.3.** Vista en la que se ha hecho el reconocimiento de la imagen objetivo.



**Figura 3.4.** Diversas vistas con las que interactúa el usuario en donde se presentan los tres sistemas anatómicos disponibles. Se observa que la posición del dispositivo móvil es indiferente siempre y cuando la imagen objetivo se encuentre enfocada



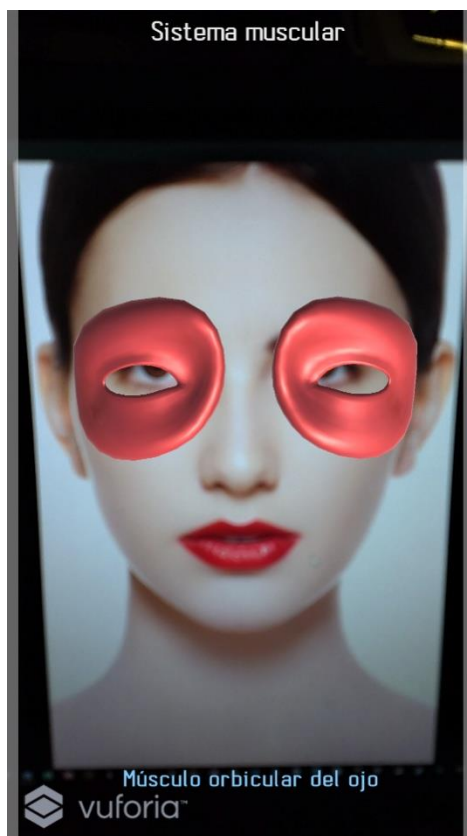
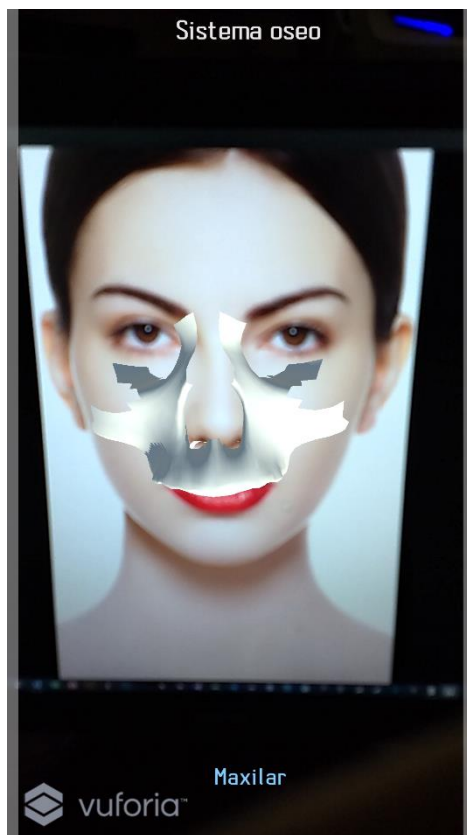


Figura 3.5. Ejemplos de selecciones de elementos de los sistemas disponibles

---

---

## Análisis y discusión de resultados

El resultado obtenido al finalizar el proyecto cumple con los requerimientos establecidos desde el planteamiento del problema, ya que la aplicación:

- Es capaz de identificar una imagen preprocesada y transformarla en un modelo tridimensional relacionado en proporción a dicha imagen.
- Permite al usuario seleccionar diversos sistemas anatómicos del modelo tridimensional así como aislar elementos de cada sistema y conocer información adicional acerca del elemento seleccionado.
- Con ayuda de la realidad aumentada permite al usuario redimensionar el modelo y observar distintas perspectivas de éste al momento de mover el dispositivo hacia la dirección deseada.
- Cuenta con una interfaz gráfica que permite al usuario realizar de forma intuitiva dichas acciones.
- Funciona de manera fluida y optimizada durante su ejecución en el dispositivo móvil.

A su vez, los módulos planteados en el diseño del sistema cumplen con la función deseada, ya que:

- El módulo de detección de rostro reconoce objetos que en efecto representan un rostro a partir de una imagen preprocesada.
- El módulo de extracción de información permite obtener toda la información relacionada a un rostro a partir de una imagen.
- El módulo de transformación es capaz de generar un modelo tridimensional a partir de datos específicos.
- El módulo de realidad aumentada agrega información a un modelo tridimensional.
- El módulo de interacción permita realizar, de forma intuitiva para el usuario, las acciones de selección, movimiento y redimensionamiento de un modelo tridimensional.

Sin embargo, es difícil afirmar que la aplicación cumple con el objetivo de mejorar el aprendizaje de la anatomía humana sin realizar antes un estudio estadístico probando la aplicación en estudiantes de ciencias de la salud que dieran una opinión clara acerca de lo que piensan del uso de esta tecnología en favor al aprendizaje. Así mismo, se requiere de la agregación de más elementos que permitan tener una aplicación final más robusta, la cual pueda ser posteriormente comercializada.



---

---

## Conclusiones

Al concluir este proyecto se cumple con los objetivos establecidos al comienzo de este, logrando obtener una aplicación para dispositivos móviles Android que, con el uso de tecnología de realidad aumentada, permite al usuario una aproximación innovadora respecto al estudio anatómico humano.

El desarrollo de este trabajo es el principio de lo que puede ser una amplia gama de aplicaciones de esta tecnología en la enseñanza de disciplinas fundamentalmente visuales, ya que esta tecnología permite una mayor inmersión del usuario y aporta elementos de entretenimiento que facilitan e incitan al estudiante a continuar con dicho estudio.

Del mismo modo, la aplicación desarrollada funge únicamente como una base del potencial que esta presenta, ya que se espera que con la futura agregación de más sistemas anatómicos, así como un mayor detalle en los modelos tridimensionales, se logre el realismo necesario para mejorar la inmersión del estudiante. Así mismo, se podría incorporar otras partes del cuerpo humano que convertirían a la aplicación en un atlas anatómico más completo. Por otro lado, para optimizar la manera en que los estudiantes adquieren el conocimiento es deseable la agregación de mecánicas como trivias y preguntas que pongan a prueba lo aprendido.

La realidad aumentada es una tecnología en constante crecimiento y en los próximos años proporcionará a los desarrolladores nuevas y más prácticas formas de aplicarla, lo cual mejorará la experiencia del usuario final y potencializará los beneficios que esta tecnología puede aportar en la rama del aprendizaje.

Es de gran importancia aprovechar las nuevas tecnologías que la ingeniería nos provee e integrarlas en otras disciplinas, lo cual permite realizar proyectos novedosos e innovadores, con el objetivo del beneficio social. La educación, siendo una de las más importantes ramas humana en la sociedad, requiere de constante innovación para facilitar a los estudiantes el aprendizaje de disciplinas complejas y esenciales para su formación como profesionistas.

---

---

## Referencias Bibliográficas

- [1] Ericsson, “*Ericsson Mobility Report*”, Stockholm, Sweden, <https://www.ericsson.com/>, 2016.
- [2] Microsoft, “*Microsoft HoloLens*”, <https://www.microsoft.com/en-us/hololens>, 2017.
- [3] Gglassday, “*Google Glass, toda la información, ¿qué es Google Glass? ¿Para qué sirven?*”, Gglassday, 2016, <http://gglassday.com/>.
- [4] J.R. Sañudo, “*The participation of Anatomy Departments in the continuing professional development of surgeons*” *European Journal of Anatomy*, vol. 11, sup. 1, pp.111-119, <http://www.eurjanat.com>, 2007.
- [5] H. E. Narváez, E. I. Murillo Rábago, “*Herramientas de estudio utilizadas por alumnos de ciencias de la salud en la materia de Anatomía.*” *Revista Investigación en Educación Médica*, Facultad de Medicina, Universidad Nacional Autónoma de México, <http://riem.facmed.unam.mx/>, 2014
- [6] Centro de Difusión de Ciencia y Tecnología, “*Realidad Aumentada*”, Instituto Politécnico Nacional. <http://www.cedicyt.ipn.mx/>.
- [7] Diccionario de la Real Academia, “*Edición del Tricentenario*”, <http://www.rae.es/>, 2017.
- [8] A.A. Ramírez Avilés, “*Sistema de detección y reconocimiento facial multi-hilo sobre GPUs*”, propuesta de proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2015.
- [9] C. Hipólito Morales, “*Cerradura electrónica con reconocimiento facial*”, propuesta de proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2014.
- [10] L.S. Fernando Vicente, “*Sistema de realidad aumentada para estrategias de mercadotecnia enfocada a una empresa inmobiliaria*”, Tesis, Facultad de Ingeniería, Universidad Nacional Autónoma de México C.U., México, 2014.
- [11] J. Cano Flórez, M. Franco Buriticá, “*Realidad aumentada aplicada a objetos de aprendizaje para asignaturas de Ingeniería Informática*”. Tesis, Facultad de Ingenierías, Politécnico Colombiano Jaime Isaza Cadavid, Medellín, Colombia, 2013.

- [12] Pharmagraphix3D, “*Pharmagraphix*”, <https://play.google.com/store/apps>, 2017.
- [13] Goodwill Enterprise, “Anatomy 3D – Anatronica”, <https://play.google.com/store/apps>, 2013.
- [14] Diana G. Oblinger, James L. Oblinger, “*Is It Age or IT: First Steps Toward Understanding the Net Generation*”, *Educating the Net Generation*, pp. 2.9, <http://www.educase.edu/educatingthenetgen/>, 2005.
- [15] P. Mendoza López, “*Lineamientos de diseño de información para el desarrollo de sitios educativos en Internet*”, Tesis profesional, Dpto. de Arquitectura y Diseño, Escuela de Ciencias Sociales, Artes y Humanidad, Universidad de las Américas Puebla, Puebla, México, 2006.
- [16] Elgin, Ben, “*Google Buys Android for Its Mobile Arsenal*”, Bloomberg Businessweek, 17 de agosto de 2005.
- [17] Jesús Tomás Gironés, “*El gran libro de Android*”, Alfaomega Grupo Editor, Segunda edición, México, 2012.
- [18] FileInfo, “*.DEX File Extension*”, <https://fileinfo.com>, 2017.
- [19] Margaret Rouse, “*Augmented Reality (AR)*”, WhatIs.com, <http://whatis.techtarget.com/>, 2016.
- [20] Pablo G. Bejerano, “*El origen de la realidad aumentada*”, Blogthinkingbig, <https://blogthinkbig.com/>, 2014.
- [21] “*Morton Heiling: The Father of Virtual Reality*”, <http://www.mortonheilig.com/>
- [22] Sutherland, I. E. “*A head-mounted three dimensional display*”, 1968.
- [23] L. B. Rosenberg. “*The Use of Virtual Fixtures As Perceptual Overlays to Enhance Operator Performance in Remote Environments.*” Technical Report AL-TR-0089, USAF Armstrong Laboratory, Wright-Patterson AFB OH, 1992.
- [24] Steven Feiner, Blair McIntyre, Doree Seligmann, “*Knowledge-Based Aumented Reality*”, Columbia University, 1993.
- [25] Unity Technologies, “*Unity3D 5.6*”, <https://unity3d.com/es/unity>, 2017.
- [26] Sergio Carlos González Sanz, “*Pokemon GO ya es la app más descargada en USA*”, ZonaRed, <https://www.zonared.com/>, 2016

- [27] Vian Gimeno, “*Realidad Aumentada Fundamentos y Aplicaciones*”, España, Universidad Politécnica de Valencia, 2011.
- [28] Virtualama, “*Realidad aumentada aplicada a la medicina*”, <http://www.virtualama.com/>, 2014.
- [29] Vuforia, “*Vuforia: Developer Library*”, <https://library.vuforia.com/>, 2017.
- [30] Richard L. Drake, Wayne Vogl, Adam W.M. Mitchell, “*Anatomía para estudiantes*”, Elsevier, España, 2005.
- [31] S. Pressman, Roger, “*Ingeniería del software: Un enfoque práctico*”, McGraw-Hill, México, 2010.
- [32] United States Patent and Trademark Office, “*Patent Full-Text Databases*”, <http://patft.uspto.gov/>, 2001.
- [33] Pinterest, <https://www.pinterest.com.mx/>, 2005.
- [34] Alicia Lam, Joyce Chan, Mavis Yao, “*Augmented and Virtual Reality Page*”, <http://newmediabusinessblog.org/>, 2016.
- [35] Anatomia-Humana, “*Todo sobre anatomía*”, <http://www.anatomia-humana.com/>, 2000.
- [36] Iván Linares, “*El giroscopio, ¿qué es y qué ocurre si mi móvil no lo tiene?*”, <https://elandroidelibre.com>, 2016.

---

---

# Apéndice A

## Código Fuente

### *swipe.cs*

```
//Script base para la acción de swipe en la pantalla táctil

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Swipe : MonoBehaviour {

    private bool tap, swipeLeft, swipeRight, slightSwipeLeft, slightSwipeRight;
    private bool isDragging = false;
    private Vector2 startTouch, swipeDelta;
    public int swipeLimit = 200;

    void Update() {

        tap = swipeLeft = swipeRight = slightSwipeRight = slightSwipeLeft = false;
        #region Entrada PC
        if (Input.GetMouseButtonDown(0)) {

            isDragging = true;
            tap = true;
            startTouch = Input.mousePosition;
        }
        else if (Input.GetMouseButtonUp(0)) {

            isDragging = false;
            Reset();
        }
        #endregion

        #region Entrada Mobil
        if (Input.touches.Length > 0) {
            if (Input.touches[0].phase == TouchPhase.Began) {
                isDragging = true;
                tap = true;
                startTouch = Input.touches[0].position;
            }
            else if (Input.touches[0].phase == TouchPhase.Ended ||
Input.touches[0].phase == TouchPhase.Canceled) {

                isDragging = false;
                Reset();
            }
        }
        #endregion
    }
}
```

```

//Calcular la distancia del swipe
swipeDelta = Vector2.zero;
if (isDragging) {
    if (Input.touches.Length > 0)
        swipeDelta = Input.touches[0].position - startTouch; //Mobil
    else if (Input.GetMouseButton(0))
        swipeDelta = (Vector2)Input.mousePosition - startTouch; //pc
}

//Se cruzo la zona del swipe?
float x = swipeDelta.x;
float y = swipeDelta.y;
if (swipeDelta.magnitude > swipeLimit) {
    //Que direccion?

    if (Mathf.Abs(x) > Mathf.Abs(y)) {
        //Izquierda o derecha
        if (x < 0)
            swipeLeft = true;
        else
            swipeRight = true;
    }

    Reset();
}
//Mover objecto ligeramente antes del swipe completo
else {
    if (Mathf.Abs(x) > Mathf.Abs(y)) {
        if (x < 0)
            slightSwipeLeft = true;
        else
            slightSwipeRight = true;
    }
}
}

private void Reset() {
    startTouch = swipeDelta = Vector2.zero;
    slightSwipeLeft = slightSwipeRight = false;
    isDragging = false;
}

public bool SlightSwipeLeft { get { return slightSwipeLeft; } }
public bool SlightSwipeRight { get { return slightSwipeRight; } }
public bool Tap { get { return tap; } }
public Vector2 SwipeDelta { get { return swipeDelta; } }
public bool SwipeLeft { get { return swipeLeft; } }
public bool SwipeRight { get { return swipeRight; } }
}

```

## swipeModel.cs

```
//Script que controla los gestos con los dedos/Movimiento entre modelos

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class SwipeModel : MonoBehaviour {

    public Swipe swipeControls;
    public RectTransform squareLeftTransform, squareRightTransform;
    public float speed;

    private bool v_tracked = false; //Es verdadero cuando un rostro se reconoce
    private int v_modelPhase = 0; //en que parte del modelo se encuentra 0-rostro
    1-musculo 2-hueso

    Text upperText;
    RectTransform upperTextTransform;

    Vector3 startPositionText;
    Vector3 startPositionSquareLeft, startPositionSquareRight;

    public int ModelPhase {
        get { return v_modelPhase; }
        set { v_modelPhase = value; }
    }

    public bool Tracked {
        get { return v_tracked; }
        set { v_tracked = value; }
    }

    void Awake() {
        m_Instance = this;
        //Instancia texto UI
        upperText = GameObject.Find("Canvas/upperText").GetComponent<Text>();
        upperTextTransform =
        GameObject.Find("Canvas/upperText").GetComponent<RectTransform>();
        startPositionText = upperTextTransform.position;

        //Instancias cuadros limite swipe
        startPositionSquareLeft = squareLeftTransform.position;
        startPositionSquareRight = squareRightTransform.position;
    }

    void Update() {

        //Comportamiento que se activa al trackear un objeto
        if (v_tracked)
        {

            Renderer[] rendererComponents =
            GetComponentInChildren<Renderer>(true);
            Collider[] colliderComponents =
            GetComponentInChildren<Collider>(true);
```

```

//Movimiento ligero del texto antes del swipe completo
if (swipeControls.SlightSwipeLeft || swipeControls.SlightSwipeRight)
{
    upperTextTransform.position = new Vector3(startPositionText.x +
swipeControls.SwipeDelta.x, startPositionText.y, startPositionText.z);
}
else
{
    upperTextTransform.position =
Vector3.MoveTowards(upperTextTransform.position, startPositionText, speed *
Time.deltaTime);
}

//Cambio de modelos
#region Modelo desplegado: Rostro
if (v_modelPhase == 0) //Estado donde se muestra el modelo del rostro
{
    if (swipeControls.SwipeLeft)
    {
        Debug.Log("Swipeo derecha");

        mostrarMusculo(rendererComponents, colliderComponents);

        upperText.text = "Sistema muscular";
        v_modelPhase = 1;
        Debug.Log("modelPhase: " + v_modelPhase);
    }
    //Se llega al limite izquierdo
    if (swipeControls.SlightSwipeRight)
        squareLeftTransform.position = new
Vector3(startPositionSquareLeft.x + swipeControls.SwipeDelta.x,
startPositionSquareLeft.y, startPositionSquareLeft.z);
    else
        squareLeftTransform.position =
Vector3.MoveTowards(squareLeftTransform.position, startPositionSquareLeft, speed *
Time.deltaTime);
}
#endregion

#region Modelo desplegado: Musculo
else if (v_modelPhase == 1) //modelo del musculo
{
    if (swipeControls.SwipeLeft)
    {
        Debug.Log("Swipeo derecha");

        mostrarHueso(rendererComponents, colliderComponents);

        upperText.text = "Sistema oseo";
        v_modelPhase = 2;
    }
    if (swipeControls.SwipeRight)
    {
        Debug.Log("Swipeo izquierda");

        mostrarRostro(rendererComponents, colliderComponents);

        upperText.text = "Sistema tegumentario";
        v_modelPhase = 0;
    }
}

```



```

    }
    #endregion

    #region Modelo desplegado: Craneo
    else if (v_modelPhase == 2) //modelo del craneo
    {
        if (swipeControls.SwipeRight)
        {
            Debug.Log("Swipeo izquierda");

            mostrarMusculo(rendererComponents, colliderComponents);

            upperText.text = "Sistema muscular";
            v_modelPhase = 1;
        }

        //Se llega al limite del lado derecho
        if (swipeControls.SlightSwipeLeft)
            squareRightTransform.position = new
Vector3(startPositionSquareRight.x + swipeControls.SwipeDelta.x,
startPositionSquareRight.y, startPositionSquareRight.z);
        else
            squareRightTransform.position =
Vector3.MoveTowards(squareRightTransform.position, startPositionSquareRight, speed
* Time.deltaTime);
    }
    #endregion
}
}

//Metodos que muestran y esconden los modelos
public void mostrarRostro(Renderer[] rendererComponents, Collider[]
colliderComponents) {
    foreach (Renderer component in rendererComponents)
    {
        if (component.tag == "Cara")
            component.enabled = true;
        else if (component.tag == "Muscle")
            component.enabled = false;
        else if (component.tag == "Hueso")
            component.enabled = false;
    }
    foreach (Collider component in colliderComponents)
    {
        if (component.tag == "Cara")
            component.enabled = true;
        else if (component.tag == "Muscle")
            component.enabled = false;
        else if (component.tag == "Hueso")
            component.enabled = false;
    }
}

public void mostrarMusculo(Renderer[] rendererComponents, Collider[]
colliderComponents) {
    foreach (Renderer component in rendererComponents)
    {
        if (component.tag == "Cara")
            component.enabled = false;
        else if (component.tag == "Muscle")
            component.enabled = true;
    }
}

```

```

        else if (component.tag == "Hueso")
            component.enabled = false;
    }
    foreach (Collider component in colliderComponents)
    {
        if (component.tag == "Cara")
            component.enabled = false;
        else if (component.tag == "Muscle")
            component.enabled = true;
        else if (component.tag == "Hueso")
            component.enabled = false;
    }
}

public void mostrarHueso(Renderer[] rendererComponents, Collider[]
colliderComponents) {
    foreach (Renderer component in rendererComponents)
    {
        if (component.tag == "Cara")
            component.enabled = false;
        else if (component.tag == "Muscle")
            component.enabled = false;
        else if (component.tag == "Hueso")
            component.enabled = true;
    }
    foreach (Collider component in colliderComponents)
    {
        if (component.tag == "Cara")
            component.enabled = false;
        else if (component.tag == "Muscle")
            component.enabled = false;
        else if (component.tag == "Hueso")
            component.enabled = true;
    }
}

//Singleton methods
private static SwipeModel m_Instance = null;
private SwipeModel() { }

public static SwipeModel Instance { get { return m_Instance; } }
}

```

## SelectElement.cs

```
//Script que controla la selección de elementos

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class SelectElement : MonoBehaviour {

    public SwipeModel mostrarModelo;
    public Swipe swipe;

    private bool aislado = false; // Indica si hay un componente aislado
    private int modelo = 0; //0-cara, 1-musculo, 2-hueso... Indica que modelo debe
    aparecer al hacer click en cualquier otro lado del objeto aislado
    private Text botText;

    void Awake() {
        botText = GameObject.Find("Canvas/botText").GetComponent<Text>();
    }

    void Update () {

        #region PC cotrols
        if (Input.GetMouseButton(0)) {
            Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
            ShootRay(ray);

        }
        #endregion

        #region mobil
        else if (Input.touchCount == 1 && Input.GetTouch(0).phase ==
        TouchPhase.Ended) {
            Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
            ShootRay(ray);
        }
        #endregion
    }

    public void ShootRay(Ray ray) {

        RaycastHit hit;
        float theDistance;

        Renderer[] renderComponents = GetComponentsInChildren<Renderer>(true);
        Collider[] colliderComponents = GetComponentsInChildren<Collider>(true);

        //Acciones a tomar cuando el raycast toque un collider, aislar componente
        if ((Physics.Raycast(ray, out hit, 10000.0f)) &&
        (swipe.SwipeDelta.magnitude == 0)) {
            theDistance = hit.distance;
            Debug.Log("Ray shot and hit");
            Debug.Log(theDistance + " " + hit.collider.gameObject.name);

            //Se recuerda cual es el modelo que esta en pantalla antes de aislar
            el componente
            if (hit.collider.gameObject.tag == "Cara")
```

```

        modelo = 0;
    else if (hit.collider.gameObject.tag == "Muscle")
        modelo = 1;
    else if (hit.collider.gameObject.tag == "Hueso")
        modelo = 2;

    //Se aísla componente
    foreach (Renderer component in rendererComponents) {
        if (component.name == hit.collider.gameObject.name)
            component.enabled = true;
        else component.enabled = false;
    }
    foreach (Collider component in colliderComponents)
    {
        if (component.name == hit.collider.gameObject.name)
            component.enabled = true;
        else component.enabled = false;
    }

    aislado = true;

    botText.text = hit.collider.gameObject.name; //Muestra el nombre del
componente aislado
    }

    //Si se hace click afuera de un collider se reaparece el modelo que habia
antes de aislar el componente
    else if (aislado == true) {
        if (modelo == 0) {
            mostrarModelo.mostrarRostro(rendererComponents,
colliderComponents);
        }
        else if (modelo == 1) {
            mostrarModelo.mostrarMusculo(rendererComponents,
colliderComponents);
        }
        else {
            mostrarModelo.mostrarHueso(rendererComponents,
colliderComponents);
        }
        aislado = false;
        botText.text = "";
    }
}
}
}

```

## DefaultTrackableEventHandler.cs

```
//Script que permite el reconocimiento de la imagen objetivo de Vuforia, con
modificaciones para el funcionamiento de la aplicación

using UnityEngine;
using UnityEngine.UI;

namespace Vuforia
{
    /// <summary>
    /// A custom handler that implements the ITrackableEventHandler interface.
    /// </summary>
    public class DefaultTrackableEventHandler : MonoBehaviour,
        ITrackableEventHandler
    {
        #region PRIVATE_MEMBER_VARIABLES

        private TrackableBehaviour mTrackableBehaviour;
        private Vector3 startPositionText;

        #endregion // PRIVATE_MEMBER_VARIABLES

        void Awake()
        {
            startPositionText =
GameObject.Find("Canvas/upperText").GetComponent<Transform>().position;
        }

        #region UNITY_MONOBEHAVIOUR_METHODS

        void Start()
        {
            mTrackableBehaviour = GetComponent<TrackableBehaviour>();
            if (mTrackableBehaviour)
            {
                mTrackableBehaviour.RegisterTrackableEventHandler(this);
            }
        }

        #endregion // UNITY_MONOBEHAVIOUR_METHODS

        #region PUBLIC_METHODS

        /// <summary>
        /// Implementation of the ITrackableEventHandler function called when the
        /// tracking state changes.
        /// </summary>
        public void OnTrackableStateChanged(
            TrackableBehaviour.Status previousStatus,
            TrackableBehaviour.Status newStatus)
        {
            if (newStatus == TrackableBehaviour.Status.DETECTED ||
                newStatus == TrackableBehaviour.Status.TRACKED ||
                newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
            {
            }
        }
    }
}
```

```

        {
            OnTrackingFound();
        }
        else
        {
            OnTrackingLost();
        }
    }

#endregion // PUBLIC_METHODS

#region PRIVATE_METHODS

private void OnTrackingFound()
{
    Renderer[] rendererComponents =
GetComponentInChildren<Renderer>(true);
    Collider[] colliderComponents =
GetComponentInChildren<Collider>(true);

    // Enable rendering:
    foreach (Renderer component in rendererComponents)
    {
        if (component.tag == "Cara") //Solo aparece la cara al momento del
trackeo
            component.enabled = true;
    }

    // Enable colliders:
    foreach (Collider component in colliderComponents)
    {
        if (component.tag == "Cara")
            component.enabled = true;
    }

    //Stuff extra
    GameObject.Find("Canvas/upperText").GetComponent<Text>().text =
"Sistema tegumentario";
    GameObject.Find("Canvas/upperText").GetComponent<Transform>().position
= startPositionText;
    SwipeModel.Instance.Tracked = true;
    SwipeModel.Instance.ModelPhase = 0;

    Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + "
found");
}

private void OnTrackingLost()
{
    Renderer[] rendererComponents =
GetComponentInChildren<Renderer>(true);
    Collider[] colliderComponents =
GetComponentInChildren<Collider>(true);

    // Disable rendering:

```

```

foreach (Renderer component in rendererComponents)
{
    component.enabled = false;
}

// Disable colliders:
foreach (Collider component in colliderComponents)
{
    component.enabled = false;
}

//Stuff extra
SwipeModel.Instance.Tracked = false;
SwipeModel.Instance.ModelPhase = 0;
GameObject.Find("Canvas/upperText").GetComponent<Text>().text = "???" ;
GameObject.Find("Canvas/upperText").GetComponent<Transform>().position
= startPositionText;

    Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " lost");
}

#endregion // PRIVATE_METHODS
}
}

```