



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**“IMPLEMENTACIÓN DE APLICACIÓN MÓVIL PARA EL  
CÁLCULO DE DOSIS DE INSULINA PARA PACIENTES  
DIABÉTICOS USANDO IMÁGENES.”**

**INFORME DE PROYECTO INTEGRADOR**

Previo a la obtención del Título de:

**INGENIERO EN COMPUTACIÓN**

**KLEBER STEVEN DIAZ COELLO**

**IVÁN MARCELO MOSQUERA CARVAJAL**

**GUAYAQUIL – ECUADOR**

**AÑO: 2018**

## **AGRADECIMIENTOS**

Agradezco primero que todo a Dios por guiarme a poder lograr este objetivo brindándome sabiduría.

Mis más sinceros agradecimientos a mis padres, Iván Mosquera Gutiérrez y Clara Carvajal Martínez, por todo el apoyo que me han dado a lo largo de mi vida, sin duda alguna ellos han sido un pilar fundamental para haber alcanzado este logro.

Les agradezco inmensamente a mis abuelitos Jaime Carvajal y Targelia Martínez porque he tenido la bendición de Dios de poder contar con ellos hasta ahora y han sido un ejemplo a seguir en mi vida.

Quiero también agradecer a mis hermanos Mario Solórzano y Jaime Mosquera por estar siempre conmigo, y apoyándome en todo momento.

Y por supuesto, quiero dar las gracias a todos los profesores que tuve a lo largo de mi carrera, por el conocimiento que me compartieron y que aplicaré a lo largo de mi vida profesional.

Iván Mosquera Carvajal

## **AGRADECIMIENTOS**

Agradezco primero a Dios por siempre estar a mi lado y darme la sabiduría para poder lograr este objetivo en mi vida. Agradezco a mis padres Kleber Diaz Medina y Vilma Coello Troya por ser los dos principales pilares en mi vida, siempre brindándome su comprensión y su amor, gracias a ellos he llegado a ser la persona que soy, además agradecer a mi hermana Valerie Diaz por ser otro pilar en mi vida, por ella he querido ser mejor para que siga mi ejemplo. Agradezco inmensamente a todos mis familiares por su cariño, a mis tíos, primos y a mis abuelos, en especial a Etelvina Troya, por siempre cuidarme. También quiero agradecer a Domenica Vera por ser la persona que me acompañó en el transcurso de mi carrera y me ayudo a superarme.

Kleber Diaz Coello

# TRIBUNAL DE EVALUACIÓN

.....  
**Mg. Rafael Rivadeneira C.**

PROFESOR TUTOR

.....  
**Ph.D. Daniel Ochoa D.**

PROFESOR COLABORADOR

## **DECLARACIÓN EXPRESA**

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

.....  
Kleber Steven Diaz Coello

.....  
Iván Marcelo Mosquera Carvajal

## RESUMEN

La diabetes tipo 1 tiene aproximadamente una población de solo el 5% de diabéticos a nivel mundial [1], siendo esta una enfermedad con una manifestación clínica que se presenta en niños y adultos jóvenes, donde las células betas del páncreas son eliminadas por el sistema inmunológico y como consecuencia no existe producción de insulina, por lo que necesitan suministrarse dosis diarias de esta hormona. Dentro de las necesidades que presenta este tipo de población está el mantener un control adecuado en las porciones de alimentos que aportan carbohidratos a su dieta [2].

Para contribuir con el control de unidades de insulina que necesitan los pacientes por la ingesta de carbohidratos al día, se desarrolló una aplicación móvil nativa para dispositivos Android, con un reconocimiento de imagen integrado, utilizando el API Vision de Google Cloud para la identificación de alimentos en un plato de comida que permitirá calcular los gramos de carbohidratos que ingerirá el paciente; a su vez los datos de los pacientes y los cálculos de insulina a suministrarse junto con los correspondientes platos de comida fueron almacenados en una base datos remota MySQL.

Palabras Clave: diabetes, insulina, reconocimiento de imagen, glucosa, carbohidratos.

## **ABSTRACT**

*Type 1 diabetes has approximately a population of only 5% of diabetics worldwide [1], this is a disease with a clinical manifestation that occurs in children and young adults, where pancreatic beta cells are eliminated by the system immunological and as a result there is no production of insulin, so they need to be given daily doses of this hormone. Among the needs presented by this type of population is maintaining adequate control in the portions of foods that provide carbohydrates to their diet [2].*

*To contribute to the control of insulin units that patients need for daily carbohydrate intake, a native mobile application was developed for Android devices. It has an integrated image recognition, using the Google Cloud Vision API for food identification in a plate of food that will allow to calculate the grams of carbohydrates that the patient will ingest; In turn, patient data and insulin calculations to be supplied along with the corresponding food dishes were stored in a remote MySQL database.*

*Keywords: diabetes, insulin, image recognition, glucose, carbohydrates.*

## ÍNDICE GENERAL

RESUMEN.....	i
<i>ABSTRACT</i> .....	ii
CAPÍTULO 1.....	1
1. DESCRIPCIÓN DEL PROBLEMA.....	1
1.1. Antecedentes.....	1
1.2. Definición del problema. ....	1
1.3. Objetivo general.....	4
1.4. Objetivos específicos. ....	4
CAPÍTULO 2.....	5
2. METODOLOGÍA DE DESARROLLO .....	5
2.1. Aplicación móvil nativa en plataforma Android.....	5
2.2. Selección de complemento para reconocimiento de imágenes.....	5
2.3. Google Cloud Vision API .....	7
2.4. Interfaz .....	9
2.5. Base de Datos .....	10
CAPÍTULO 3.....	12
3. IMPLEMENTACIÓN.....	12
3.1. Diagrama de despliegue .....	12
3.2. Módulo de autenticación .....	13
3.3. Módulo de cálculos de dosis de insulina .....	17
3.2.1 Modo Manual para el cálculo de dosis de insulina.....	18
3.2.2 Modo Automático para el cálculo de dosis de insulina.....	21
3.4. Módulo de historial.....	22
CAPÍTULO 4.....	24
4. RESULTADOS.....	24
4.1. Pruebas de usabilidad .....	24
4.2. Prueba de efectividad en el reconocimiento de platos de comida. ...	25



CONCLUSIONES Y RECOMENDACIONES .....	28
BIBLIOGRAFÍA.....	29

# **CAPÍTULO 1**

## **1. DESCRIPCIÓN DEL PROBLEMA.**

### **1.1. Antecedentes.**

Los pacientes de FUVIDA son personas que poseen la enfermedad Diabetes Mellitus tipo 1, que se caracteriza por el aumento de los niveles de glucosa en sangre (hiperglucemia) a causa de la disminución de la formación de una hormona llamada insulina por parte de las células beta del páncreas [3]. Estos pacientes necesitan un manejo diario y personalizado de su glucosa sanguínea, por lo que deben ajustar sus dosis de manera constante, de acuerdo a sus comidas, nivel de actividad física y estado fisiológico. Este ajuste requiere que el niño o sus padres realicen cálculos matemáticos relativamente simples que consideren las variables antes enunciadas, para que la dosis sea útil y eficaz.

### **1.2. Definición del problema.**

FUVIDA necesita una herramienta tecnológica que facilite al paciente o sus familiares estos cálculos tomando en cuenta los aspectos destacados anteriormente, que sea de bajo costo, portable, de rápida implementación y fácil uso.

Para el manejo de la enfermedad, se han desarrollado una serie de soluciones, en especial para el Diabetes Mellitus 1, por ejemplo, las bombas de insulina. Estos equipos han ido evolucionando a lo largo del tiempo y hoy en día son microcomputadores, capaces de medir glucosa e inyectar insulina mediante una programación establecida. El inconveniente con estos dispositivos es que son extremadamente caros (entre \$5000 a \$10000) [4], por lo que se encuentran alejados de la

realidad económica de FUVIDA. Además, su manejo requiere de cierta experticia.

Actualmente, hay la posibilidad de encontrar en Internet sitios gratuitos donde se puede calcular la dosis diaria de insulina considerando el nivel de glucosa y la ingesta de carbohidratos. En estos casos, hay algunas limitantes, como la necesidad del pago mensual o anual para el uso ciertas aplicaciones móviles y la no consideración de muchos platos típicos de la gastronomía de nuestra región o país. Por lo que el cálculo de insulina es más complicado e inexacto al no poder realizar la conversión exacta de carbohidratos a partir de la dieta local.

A continuación, como se muestra en la Tabla 1 se da a conocer una comparación de la aplicación móvil desarrollada (LINA) con otras aplicaciones ya existentes en el mercado.

Nombre	Idioma Español	Extensa gama de alimentos	Platos Típicos ecuatorianos	Reconocimiento de la comida por medio de fotos	Cálculo de insulina	Cálculo de insulina Optimizado
LINA	X	X	X	X	X	X
Insucontrol	X		X		X	
diabetesalacarta	X	X				
MydiabeticAlert		X				
Otros		X				

**Tabla 1.1 - Comparación entre aplicaciones**

Por estas razones, se desarrolló una aplicación móvil para dispositivos Android, que permite calcular la dosis de insulina a partir de la dieta diaria

del paciente, considerando los platos típicos de la gastronomía ecuatoriana y sus valores diarios de glucosa. El paciente puede seleccionar manualmente el plato a ingerir de una lista de platos típicos de Ecuador. Luego, la aplicación realiza internamente la conversión de la comida ingerida en gramos de carbohidratos, después estos valores son transformados en unidades de insulina. Estos valores de insulina son ajustados en función de las actividades físicas del paciente y de su estado fisiológico previo. Finalmente, la aplicación tiene un valor agregado adicional, tal como se muestra en la Figura 1.1, el cual consiste en la posibilidad de que el paciente capture una foto al plato de comida a ingerir y este sea reconocido mediante el uso de redes neuronales [5] [6] [7], identificando los diferentes alimentos que contiene tal plato, para que luego el paciente ingrese manualmente las porciones o cantidades que va a ingerir de dichos alimentos reconocidos y como resultado obtenga la dosis de insulina adecuada a inyectarse.



**Figura 1.1 - Diagrama de funcionamiento de la aplicación**

Los valores de glucosa diarios ingresados más la información de los alimentos ingeridos, transformados en gramos de carbohidratos, son registrados en una base de datos en línea.

### **1.3. Objetivo general.**

Crear una aplicación móvil Android para el cálculo de la dosis diaria de insulina para pacientes con diabetes mellitus tipo 1, basado en el reconocimiento visual de alimentos provenientes de la dieta de la población ecuatoriana; siendo el cálculo de insulina fundamentado en los factores biopsicosociales de los pacientes que se ajustan por un modelo matemático con una interfaz amigable empaticada con el usuario final.

### **1.4. Objetivos específicos.**

- Integrar un reconocimiento de imagen para la identificación de alimentos en un plato de comida que permitirá calcular los gramos de carbohidratos que ingerirá el paciente luego de que ingrese manualmente las porciones o cantidades de cada alimento reconocido.
- Elaborar una base de datos con los más importantes platos típicos ecuatorianos junto con su cantidad de gramos de carbohidratos por porción.

## CAPÍTULO 2

### 2. METODOLOGÍA DE DESARROLLO

#### 2.1. Aplicación móvil nativa en plataforma Android

Para el desarrollo de la aplicación móvil se decidió que sea nativa Android, por lo que la misma tuvo que ser desarrollada en el lenguaje de programación Java. Entre las ventajas de que sea nativa se tiene que existe un máximo rendimiento en los dispositivos, una mejor experiencia por parte del usuario y la posibilidad de acceder a todas las características del hardware del dispositivo: cámara, GPS, dispositivos de almacenamiento, entre otras. En este caso se requirió acceder a la cámara del teléfono móvil debido que se la necesita para tomar la foto a los platos de comida.

#### 2.2. Selección de complemento para reconocimiento de imágenes

Como se comenta en el capítulo anterior, se incorporó un reconocimiento de imágenes en la aplicación, el cual funciona mediante redes neuronales, que son una simulación de cómo funciona un cerebro típico humano en una computadora, es decir, que pueden aprender, detectar patrones y tomar decisiones en torno a ello, tal y como lo haría un ser humano. Una red neuronal toma una enorme cantidad de información previa, conocido como “ejemplos de entrenamiento” y luego utiliza esta información para inferir una respuesta por su cuenta. [8]

Por ello se realizaron pruebas entre *Google Cloud API Vision* y *Computer Vision API de Microsoft Azure*, que ya tienen encapsuladas en sí potentes modelos de aprendizaje automático.

Se comparó el porcentaje de efectividad de reconocimiento entre ambas API's [9], el cual fue definido como el porcentaje en identificar sustantivos correctamente de una imagen.

Foto	Sustantivos totales	Sustantivos reconocidos Google Vision	Sustantivos reconocidos Microsoft Azure
Perro	1	1	1
Ensalada de Frutas	3	2	1
Arroz con pescado frito y ensalada	3	2	1
Comedor	2	2	2
Seco de gallina	3	2	2
Laptop	1	1	1
Pelota de fútbol	1	1	1
Paisaje	3	3	2
Carro	1	1	1
Vestimenta	3	2	1
Total porcentaje efectividad Google API Vision: <b>86.66%</b> Total porcentaje efectividad Microsoft Azure Vision API: <b>73.33%</b>			

**Tabla 2.1 – Tabla que indica la tasa de efectividad de reconocimiento de sustantivos entre el API Vision de Google y el Microsoft Vision API.**

Como resultado se obtuvo que el modelo de aprendizaje automático de Google Cloud API Vision supera en efectividad al de Computer Vision API de Microsoft al reconocer con etiquetas claves la foto tomada.

Así mismo, se comparó el costo de usar ambas API's. Se encontró lo siguiente:

- Microsoft da más transacciones gratis, 5000 vs las 1000 que da Google.
- El costo en Microsoft a partir de las 5000 hasta 1 millón de transacciones

es de \$1 por cada 1000 fotos y luego del millón en adelante es de 0.80\$ por cada 100 fotos.

- En Google, el costo a partir de la transacción 1001 hasta las 5 millones de transacciones es de \$1.50 por cada 1000 fotos al mes, y luego de las 5 millones en adelante es de \$1 por cada 1000 fotos al mes.

Por ello, luego del análisis y comparaciones realizadas a ambas API's, a pesar de que el costo de Computer Vision API de Microsoft era menor, se decidió usar el API Vision de Google Cloud [6] como modelo de aprendizaje automático para el reconocimiento de platos de la aplicación debido a que, como ya se demostró, supera en gran porcentaje de efectividad en reconocimiento al API de Microsoft.

### **2.3. Google Cloud Vision API**

La API de Google Cloud Vision permite a los desarrolladores integrar fácilmente funciones de detección de visión dentro de las aplicaciones, es decir, funciones que logren identificar los objetos que hay dentro de una imagen, incluido el etiquetado de imágenes, detección de rostro y punto de referencia, reconocimiento óptico de caracteres (OCR) y etiquetado de contenido explícito.

Esta API tiene como marco interno de aprendizaje automático a TensorFlow, que es una biblioteca de código abierto que se basa en un sistema de redes neuronales [10], en donde Google utiliza como datos para entrenar a su red, todas esas millones de fotos que los usuarios en el mundo almacenan en los servicios en la nube que Google ofrece gratuitamente como *Google Fotos*.



La API permite que los desarrolladores comprendan el contenido de una imagen mediante el encapsulado de potentes modelos de aprendizaje automático en una API REST fácil de usar. Es capaz de clasificar imágenes rápidamente en miles de categorías (por ejemplo, "avión", "tigre" o "Estatua de la Libertad"), detecta objetos como flores, animales, medios de transporte, platos de comida o miles de otras categorías de objetos habituales, así como también reconoce caras individuales dentro de las imágenes, además de buscar y leer palabras impresas en ellas. Puede crear datos que describen el contenido de los archivos o la información de los mismos en un catálogo de imágenes, moderar el contenido ofensivo o habilitar nuevas situaciones de marketing mediante el análisis de sentimientos en imágenes. Tiene la posibilidad de analizar las imágenes cargadas en la solicitud o integrarlas en un sistema de almacenamiento de imágenes de Google Cloud Storage. Cabe recalcar que la API Vision mejora con el tiempo, ya que se añaden conceptos nuevos y se aumenta la precisión [8].

Para el desarrollo de esta aplicación se usó en particular el '*Label Detection*', que es una de las tantas características del API Vision de Google. Esta característica consiste en detectar amplios conjuntos de categorías dentro de una imagen, es decir, agrega semántica a una imagen al proporcionar un conjunto de etiquetas relevantes o palabras claves.

Dicha funcionalidad fue aplicada en la modalidad automática de cálculo de dosis de insulina de la aplicación, ya que el usuario al tomar la foto al plato de comida recibe como respuesta del API en la pantalla las etiquetas de alimentos detectados de dicha imagen, en donde previamente antes de mostrarse en pantalla se filtran y eliminan aquellas etiquetas muy generales o irrelevantes que no describan de manera precisa a los alimentos del plato. Posterior a aquello, el usuario será

capaz de eliminar las etiquetas de alimentos que no describen el plato deslizándolas en la pantalla hacia la izquierda si solamente las desea descartar y a la derecha si desea eliminarlas con el objetivo de que no se vuelvan a mostrar en un próximo reconocimiento de una foto. De esta manera, mientras más sean los usuarios que deslizen hacia la derecha una etiqueta irrelevante, la aplicación será inteligente ya que irá tomando en cuenta aquello y llegará a un punto en donde no mostrará dichas etiquetas a ningún usuario en los próximos reconocimientos de fotos. Además, en el caso de que exista uno o más alimentos que el API no logre reconocer en la foto tomada, el usuario tiene la posibilidad de poder ingresar manualmente dichos platos faltantes desde una lista, indicando la cantidad de porciones para cada plato de comida seleccionado. Finalmente, el usuario procede a generar el cálculo de dosis de insulina correspondiente.

#### **2.4. Interfaz**

Para la interfaz se decidió usar una combinación de colores que generen comodidad y una sensación agradable para el usuario. Por ello, se eligieron los colores celeste, que representa a la diabetes, y el color blanco que está relacionado directamente con la pureza y confiabilidad, que son características que van de la mano con el concepto de la aplicación. Así mismo se decidió hacer uso de un logo que lo simboliza. Su significado es muy positivo, en muchas culturas. El color celeste representa el cielo que une a todas las naciones y es el color de la bandera de Naciones Unidas [11].

Se hizo énfasis en la usabilidad de la aplicación [12], interfaces sencillas, con textos legibles, iconos e imágenes fáciles de comprender, para no llegar a generar incomodidad o confusión en el usuario.

## 2.5. Base de Datos

Como se puede ver en la Figura 2.1, la base de datos consta de 5 tablas: Usuario, Categoría, Comida, Historial, Detalle\_Comida ,Labels\_No\_Necesarios.

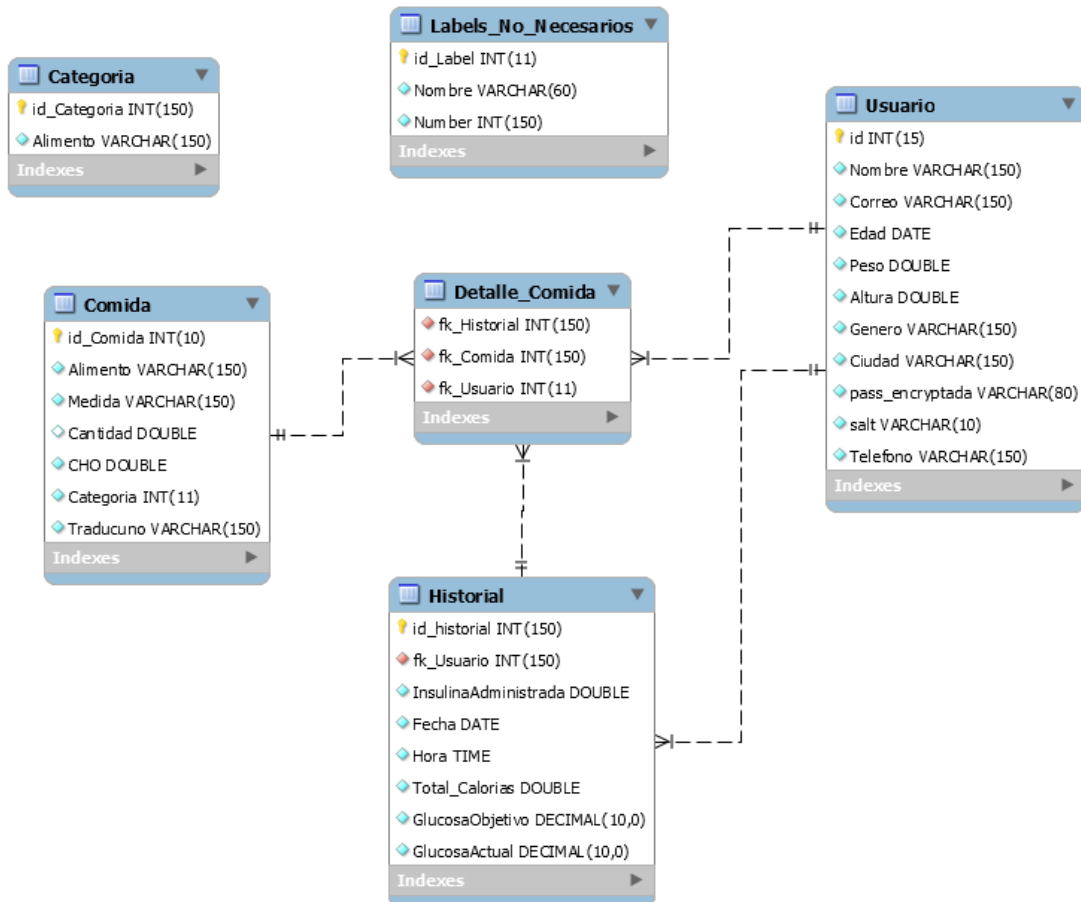


Figura 2.1 - Diagrama Base de Datos LINA

Para la creación de la base de datos de la aplicación, la cual es relacional, se escogió MySQL, que tiene como ventajas ser un software Open Source, con alto rendimiento, conectividad y seguridad [13]. La base se encuentra almacenada de forma remota.

Los datos que se encuentran guardados dentro de la base son los diferentes platos de comidas típicas del Ecuador junto con su

equivalencia en gramos de carbohidrato por taza o porción, además de un campo *Traducción* dentro de la tabla *Comida* que sirve para traducir las etiquetas en idioma inglés que obtiene la aplicación luego de realizar el reconocimiento a una foto tomada. Así mismo, se guarda con fecha y hora un historial de los platos de comida que ingiere el usuario, junto con el total de gramos de carbohidratos y la insulina suministrada. Dicho historial por cada usuario servirá para que en un futuro se pueda realizar una página web donde el doctor pueda llevar un seguimiento de todo lo que sus pacientes están consumiendo, además de variables como su peso, edad, estatura, entre otras.

Por otro lado, la tabla *Labels\_No\_Necesarios* contiene almacenadas todas las etiquetas irrelevantes que los usuarios eliminan al deslizarlas a la derecha luego de haber tomado foto a un plato de comida y que posteriormente, cuando su contador de eliminaciones llegue a veinte, se filtran cuando son detectadas en reconocimientos futuros.

# CAPÍTULO 3

## 3. IMPLEMENTACIÓN.

### 3.1. Diagrama de despliegue

La aplicación fue implementada en Android Studio versión 3.0.1, y estará en constante interacción con los servidores de Google al utilizar el API Vision y el servidor web que aloja la base de datos en MySQL. Además existe una base de datos interna SQLITE que sirve para consultar los platos de comida cuando no haya conexión a internet.

En la Figura 3.1 se muestra dicha distribución de los procesos y componentes y cómo colaboran entre sí:

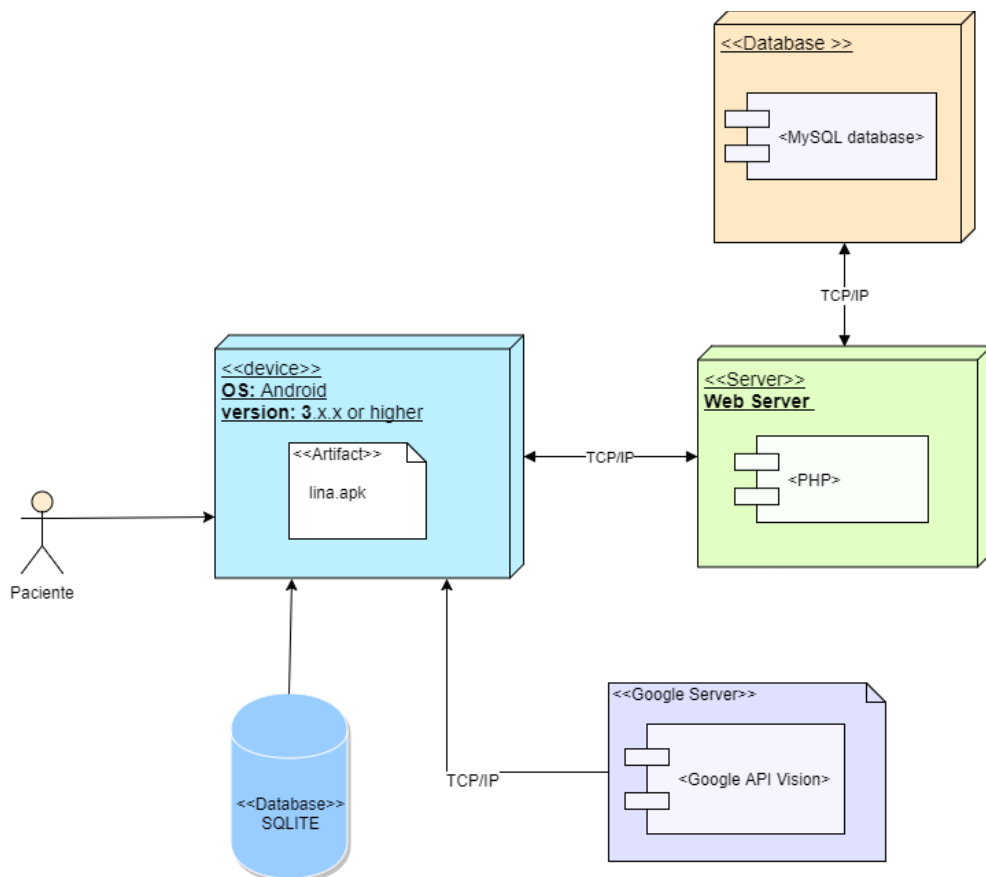


Figura 3.1 - Diagrama de despliegue de la aplicación

Para la implementación se definieron los siguientes módulos:

- Módulo de autenticación
- Módulo de cálculo de dosis de insulina
- Módulo de historial

Como se puede observar en la Figura 3.2 se muestra el diagrama de bloques de la aplicación, que es una representación gráfica de su funcionamiento y cómo se relacionan todos los módulos.

### **3.2. Módulo de autenticación**

En este módulo el paciente tiene la posibilidad de autenticarse con su correo electrónico y contraseña para acceder a las funcionalidades de la aplicación como muestra la Figura 3.3. En el caso que el paciente no se encuentre registrado deberá crear una cuenta ingresando sus correspondientes datos tal como se ve en la Figuras 3.4, 3.5 y 3.6 en donde se usó el componente *EditText* para permitir dicha entrada de datos. Existe la limitante de que el usuario necesita tener conexión a internet para poder iniciar sesión, debido a que la base de datos interna solo contiene almacenados los platos de comida para que el usuario tenga la posibilidad de utilizar el cálculo de modo manual sin conexión a internet pero solo en el caso que ya anteriormente se haya autenticado con conexión a internet.

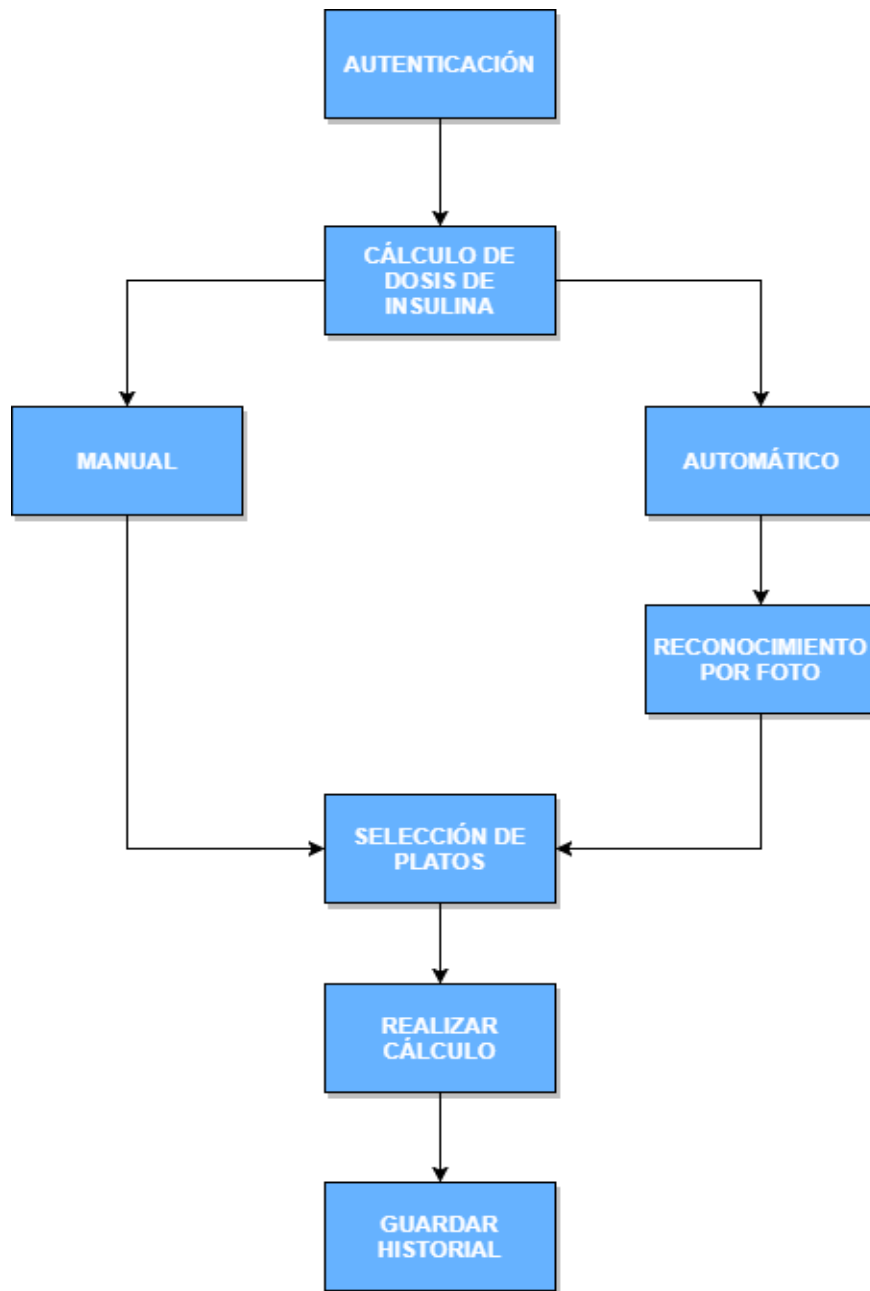
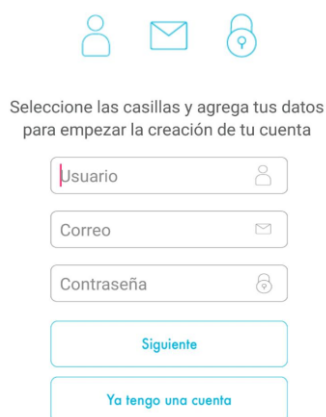


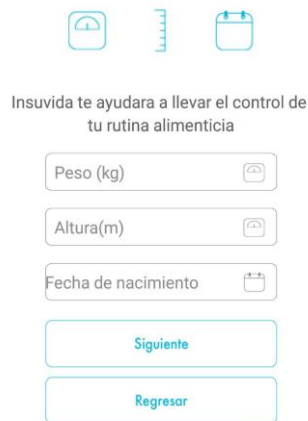
Figura 3.2 - Diagrama de bloques de la aplicación



**Figura 3.3**



**Figura 3.4**



**Figura 3.5**



**Figura 3.6**

Al iniciar sesión se muestra el perfil del paciente con sus datos personales ingresados como se puede ver en la Figura 3.7. Dicha sesión se



mantendrá activa mientras el usuario no cierre la misma. Además el paciente tiene la posibilidad de editar sus datos si así lo desea.

Por otro lado, se muestra una barra inferior, que fue implementada con el uso del componente de Android Studio llamado Navigation View. Esta barra de menú cuenta con las diferentes opciones en las cuales el usuario puede navegar: tal como Perfil, Cálculo de Insulina, los Platos y por último el Historial de los cálculos realizados. En la siguiente sección se explica con más detalle qué funcionalidades cumplen cada uno de estos módulos.



Figura 3.7

### 3.3. Módulo de cálculos de dosis de insulina

Es el módulo más complejo de la aplicación ya que consta de dos modos para su funcionamiento como se observa en la Figura 3.8. El primer modo, llamado *Modo Manual*, consiste en realizar el cálculo de dosis de insulina luego de que el paciente ingrese manualmente el plato que va a ingerir y su respectivo número de porciones. Mientras que el *Modo Automático* consiste en realizar el cálculo de dosis de insulina a partir del reconocimiento automático de los alimentos mediante la toma de una foto.

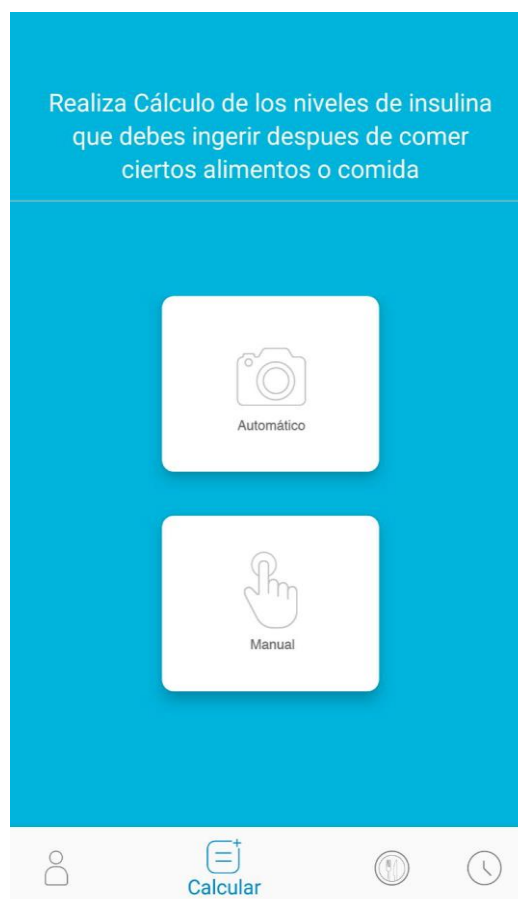


Figura 3.8 Layout de selección de modo de cálculo de dosis de insulina

### 3.2.1 Modo Manual para el cálculo de dosis de insulina

Al escoger el *Modo Manual* se pide al paciente que realice una búsqueda del plato de comida que desee al seleccionar el mismo desde una lista donde le indica los gramos de carbohidrato por taza o porción y que además tiene implementado un filtro para ayuda de búsqueda del plato, tal como se puede apreciar en la Figura 3.9. Adicionalmente por cada plato de comida que el paciente seleccione debe indicar el número de tazas o porciones. Dichos datos mostrados en la lista son extraídos desde la base de datos remota, haciendo una petición con el método POST, dando como resultado un archivo JSON, el cual es leído y extraído para así agregarlo y que sea visible para el usuario usando el componente RecyclerView.

Platos	Gramos de CHO
Ensalada de verduras 1/2 plato	2.11
Ensalada de veteraba 3 cucharadas	5.53
Ensalada rusa 2 cucharadas	8.41
Ensalada César 1/2 plato	4.3
Ceviche de pescado 1 taza	5.85

Figura 3.9 Layout de selección de platos

Luego el paciente debe también ingresar otros datos que son: glucosa actual, glucosa objetivo y factor de sensibilidad (ICR) que son necesarios para realizar el cálculo. Ver Figura 3.10.

The screenshot shows the 'Cálculo de Insulina' app interface. At the top, there is a blue header with a back arrow and the text 'Cálculo de Insulina'. Below the header, there are four input fields with labels and values: 'Nivel Glucosa Actual : 0', 'Nivel Glucosa objetivo : 0', 'Factor de sensibilidad (ICR) : 15', and 'Dosis diaria de insulina (DDI) : 0'. Below these fields, there is a section for 'Total de Carbohidratos : 0 g' with a button labeled 'Agregar plato'. A note below the button reads: 'Nota: Recuerde que esta aplicación solo considera el uso de insulina ultrarapida para el cálculo de la dosis.' At the bottom, there is a navigation bar with four icons: a person icon, a 'Calcular' button with a plus sign, a circular arrow icon, and a clock icon.

**Figura 3.10 Modo manual de cálculo de dosis de insulina**

En el momento que el usuario va agregando los platos, estos son guardados dentro de una base interna SQLite de la aplicación, lo que permite al usuario poder usar sin conexión a internet este modo Manual. Luego, de aquella base de datos se extraen los datos y se incluyen dentro de una lista de solo las comidas que el usuario haya seleccionado, en donde también tiene la posibilidad de eliminar el plato en el caso que se haya equivocado o ya no lo desee comer.

La aplicación va mostrando la cantidad de carbohidratos totales cada vez que agregue o elimine un plato de comida.

Posterior a que el usuario haya ingresado todos los datos requeridos y haya seleccionado los platos de comida, se calcula el total de insulina a inyectar. La fórmula usada para el cálculo es reconocida a nivel mundial por la mayoría de endocrinólogos y asociaciones de diabetes [14]:

$$BI = (CHO/ICR) + ((NGActual - NGObjetivo)/FC) \quad (3.1)$$

*BI*:Bolos de Insulina.

*CHO*:Carbohidratos ingeridos, debe ser seleccionado de la base de datos de carbohidratos según la comida que ingiera el paciente.

*ICR*: Factor de sensibilidad, es la cantidad de CHO necesario para que coincida con el efecto reductor de BG (glucosa en sangre pre calórica) de 1 unidad de insulina. Debe ser ingresado por el propio paciente.

*NGActual*: Nivel de Glucosa Actual.

*NGObjetivo*: Nivel de Glucosa Objetivo,

Deben ser ingresados por el propio paciente. Debido a que cada nivel de glucosa en la sangre de los pacientes es diferente y cada uno tiene un nivel objetivo de glucosa en la sangre diferente también.

*FC*: Factor de Corrección, sirve para que el paciente ajuste su glucosa dependiendo del tipo de insulina que vaya a suministrarse. En el caso de esta aplicación debido a que gran porcentaje de los pacientes de FUVIDA utilizan insulina ultrarrápida, el factor de corrección está definido como el cociente entre 1800 y el número de dosis diarias de insulina que ingrese manualmente.

Esta fórmula (3.1) tiene una tasa del 99% de éxito biológico al estar basada en el método de Conteo de Carbohidratos para que permita realizar el correcto cálculo de bolo de insulina en los pacientes de diabetes mellitus tipo 1.

Luego de realizar el cálculo se muestra las unidades de insulina a inyectar. Además, el usuario tiene la opción de guardar la consulta en la base de datos para que luego pueda ver los alimentos que seleccionó y las unidades de insulina en la opción de historial. También tiene la opción de hacer un nuevo cálculo.

### **3.2.2 Modo Automático para el cálculo de dosis de insulina**

En lo que respecta al modo automático, se le permite al paciente la captura de una foto al plato de comida, tal como se muestra en la Figura 3.11.

Luego de que el paciente tome la foto, la aplicación hace un requerimiento al API Vision de Google, que retorna como respuesta todas las etiquetas reconocidas de dicha foto. Y luego, tal como se mencionó en el capítulo#2 se eliminan aquellas palabras o etiquetas que son innecesarias. Las elimina comparándolas con las etiquetas que han sido incluidas en la tabla '*Labels\_No\_Necesarios*' dentro de la base de datos remota, es decir, si en la respuesta hay etiquetas que existen en dicha tabla entonces estas son eliminadas. Luego, las etiquetas que no fueron eliminadas son presentadas en una lista, las cuales el paciente puede decidir si alguna es un ingrediente que está en el plato. Para lo cual tiene dos opciones, ignorar las etiquetas que no describan al plato deslizándolas hacia la izquierda, ó, deslizarlas hacia la derecha con el propósito de que dichas etiquetas no vuelvan a aparecer en futuros reconocimientos como se explica en el capítulo#2. Luego de que esté todo bien la aplicación redirige al usuario a la pantalla de cálculo manual

Figura 3.10, pero esta vez ya con los ingredientes reconocidos, en donde podrá realizar lo mismo que se puede hacer en el modo manual.

Tomar la fotografía ayuda de tal forma de forma automática los alimentos y bebidas que estas por ingerir



**Figura 3.11 Modo automático de cálculo de dosis de insulina.**

### **3.4. Módulo de historial**

En el módulo de Historial el paciente podrá ver todos los cálculos que ha guardado, los cuales mostrará la fecha, hora, el total de

carbohidratos consumidos, el total de insulina suministrada, la glucosa actual y la glucosa objetivo. Ver Figura 3.12



Figura 3.12 Layout de historial de la aplicación



## **CAPÍTULO 4**

### **4. RESULTADOS.**

Se realizaron dos tipos de pruebas: pruebas de usabilidad y pruebas de efectividad en el reconocimiento de alimentos.

#### **4.1. Pruebas de usabilidad**

Se procedió a probar la aplicación en cuatro pacientes de FUVIDA que padecen de diabetes tipo uno: dos adolescentes y dos adultos. En donde se analizaron atributos como: nivel de dificultad en usar la aplicación y el diseño de la interfaz de usuario.

El atributo de nivel de dificultad en usar la aplicación se lo calificó en una escala de tres valores posibles: Fácil de comprender, Normal de comprender y Difícil de comprender, en donde tres de los cuatro pacientes coincidieron que la aplicación es Normal de comprender, ya que cuenta con una adecuada distribución de los elementos mostrados y textos legibles. Mientras que el otro paciente indicó que le pareció Fácil comprender el uso de la aplicación.

Con respecto al atributo de satisfacción del usuario con el diseño de la interfaz se decidió calificarlo en una escala con los valores posibles de: No me agrada y Me agrada. Los dos pacientes adolescentes coincidieron que no les agrada porque consideran que la interfaz es muy simple y con colores no saturados. Ellos sugieren una interfaz más interactiva y más entretenida. En cambio los dos pacientes adultos indicaron que sí les agrada la interfaz ya que la ven acorde a una aplicación confiable con fines médicos, que es su principal motivo de uso.

#### 4.2. Prueba de efectividad en el reconocimiento de platos de comida.

Se la realizó para conocer la tasa de efectividad de reconocimiento de platos del API Vision de Google. Dicho porcentaje fue definido como el cociente entre los alimentos reconocidos por el API y los alimentos totales en el plato. Se seleccionaron veinte diferentes platos y bebidas que son consumidos habitualmente en Ecuador como se muestra en la Tabla 4.1:

Plato o bebida	Alimentos totales	Alimentos reconocidos Google Vision	Porcentaje de efectividad Google Vision
Humita	1	0	0
Guatita	3	2	66
Arroz con pescado frito y ensalada	3	2	66
Arroz con menestra y carne	3	2	66
Seco de gallina	3	2	66
Encebollado	4	1	25
Llapingacho	4	2	50
Arroz con encebollado de atún	3	1	33
Limonada	1	1	100
Tostada de jamón y queso	1	1	100
Pan blanco	1	1	100
Caldo de gallina	2	2	100
Arroz con puré y carne apanada	3	1	33
Lasaña	1	1	100
Tallarín	1	1	100
Estofado de carne y arroz	2	2	100
Helado de chocolate	1	1	100
Empanada	1	1	100
Bollo de pescado con arroz	2	1	50
Churrasco	4	2	50
Total porcentaje efectividad Google API Vision: <b>61.36%</b>			

**Tabla 4.1 – Tabla que indica la tasa de efectividad de reconocimiento de platos del API Vision de Google.**

A continuación se muestran dos de las fotos de los platos de comida que fueron reconocidos en su totalidad por el API: Figura 4.1 y Figura 4.2.

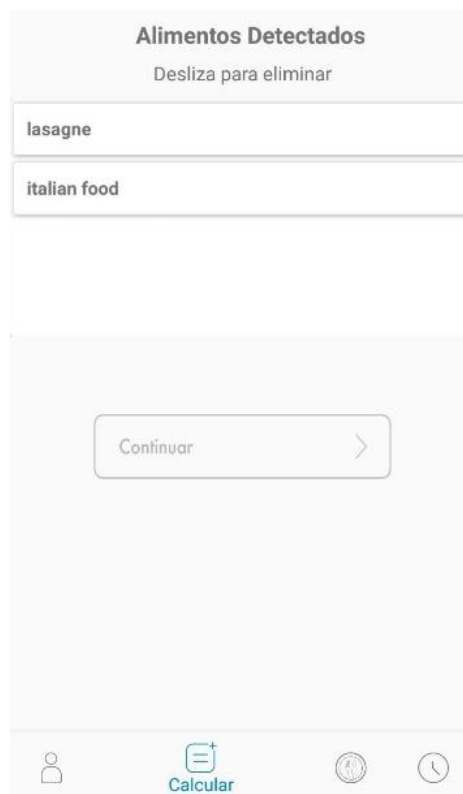


**Figura 4.1 Estofado de carne**



**Figura 4.2 Lasaña**

En el plato del estofado de carne el API logró reconocer los dos alimentos que contiene: arroz y estofado. Lo mismo ocurre en el otro plato mostrado, cuyos resultados se muestran a continuación en la Figura 4.3:



**Figura 4.3 Etiquetas reconocidas de la Figura 4.2**

En cambio, en los platos mostrados en la Figuras 4.4 y Figura 4.5, hubo muy poca efectividad por parte del API al reconocer los alimentos. Y esto ocurre debido a que son platos únicamente consumidos en Ecuador, por lo tanto, el motor de aprendizaje de Google Vision API aún no logra detectar, o también ocurre porque en muchos casos ciertos alimentos están ocultos en la parte inferior del plato de comida y ni el ojo humano es capaz de detectarlos a simple vista.



**Figura 4.4 Humita**



**Figura 4.5 Encebollado**

Como resultado de la prueba se obtuvo que el modelo de aprendizaje automático de Google Cloud API Vision tiene una efectividad del 61.36% en reconocer los alimentos de un determinado plato que tienen como característica principal el hecho de ser habitualmente consumidos en Ecuador. Cabe recalcar que al tratarse de una red neuronal este porcentaje tiene la posibilidad de incrementarse si se realiza un entrenamiento a la red con los platos típicos ecuatorianos, que en esta ocasión, por motivos de tiempo requerido en dicho entrenamiento, no pudo realizarse.

## CONCLUSIONES Y RECOMENDACIONES

En la aplicación se realizó lo que a principio se acordó con el cliente, donde se incluyó que tenga una adaptabilidad en cada uno de los usuarios, con un fácil manejo e interacción; los pacientes de FUVIDA tienen una forma manual de calcular mediante conteo de carbohidratos y subsecuentemente las dosis de insulina a suministrarse, pero con la creación de la aplicación móvil tendrán una forma más rápida de calcular la insulina que deben suministrarse, y además como todos los datos son guardados en una base de datos remota, estos servirán para en un futuro por medio de una página web, el médico de los pacientes pueda tener un mejor seguimiento de cada uno de ellos.

Con los datos obtenidos se puede realizar una mejora del cálculo de unidades de insulina, ya que al contar con los datos de medición, posteriormente se pueden hacer correcciones con el uso de variables estadísticas para poder llegar a un nivel óptimo, que comúnmente no han sido tomadas en cuenta al momento de hacer mediciones como, actividad física, sexo, entre otros.

Para futuros usos de aplicaciones con reconocimiento se recomienda hacer una exhaustiva investigación para saber qué API es más certera. En el caso del reconocimiento de platos fue la de Google Vision, pero puede que en otro tipo de reconocimiento como de caras o letras sea mejor otra API.

## BIBLIOGRAFÍA

- [1] American Diabetes Association (ADA). (2014). Type 1 Diabetes.
- [2] Federación Internacional de la Diabetes (FID). (2015, Noviembre 12). Atlas de la Diabetes de la FID.
- [3] Diabetes Tipo 1 | FUVIDA, Disponible en: <http://www.fuvida.org.ec/diabetes-tipo-1/>
- [4] Bombas de insulina, una opción que debes conocer, Disponible en: <http://www.jediazucarado.com/bombas-de-insulina-una-opcion-disponible-que-debes-conocer/>
- [5] P. Pablo García, Reconocimiento De Imágenes Utilizando Redes Neuronales Artificiales. España: Facultad De Informática, Universidad Complutense De Madrid, 2013.
- [6] API de Google Cloud Vision. Disponible en: <https://cloud.google.com/vision/?hl=es>
- [7] Computer Vision Api - Microsoft Azure. Disponible en: <https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/>
- [8] A. Goga. ¿Qué son las Redes Neuronales? Cómo funcionan y para qué sirven?. Disponible en: <https://www.arturogoga.com/redes-neuronales-que-son-y-como-funcionan/>
- [9] ¿Qué es una API y para qué sirven?. Disponible en: <http://www.abc.es/tecnologia/consultorio/20150216/abci--201502132105.html>
- [10] TensorFlow. Disponible en: <https://www.tensorflow.org/>
- [11] Diabetes, F. (2017). Fundación para la Diabetes. [online] Fundaciondiabetes.org. Disponible en: <http://www.fundaciondiabetes.org/>

[12] M. L. A. Domínguez. (2010). ¿Qué es usabilidad?. Disponible en: [https://deb5b5b3-a-62cb3a1a-sites.googlegroups.com/site/caudalstudio/home/%C2%BFqueesusabilidad/Usabilidad\\_2010.pdf?attachauth=ANoY7cpboTgegk0jFpmofkXSt0r5XS\\_66GlxE DzDbzPAPkgh38POZn\\_tRc6so7O\\_rNv0yudprXARvyGzYCAGJEYBWqPabc BjpOC\\_kj\\_ugNBbINX6xFgObc8YZfu0niPwSPZVPveurgX7-rSkqGdQBoXAHBjpt1Y5rpMj0tb5xfZ\\_IGW3Bq0AUEbMDZVzF7zX\\_Ls5QBc wKZkSRAUjXg8RVzONq3X- jf97kzg9tllzj7Ql2afr04NwW\\_ZJ6ABVPhwrM5eIH-KDwlOR&attredirects=0](https://deb5b5b3-a-62cb3a1a-sites.googlegroups.com/site/caudalstudio/home/%C2%BFqueesusabilidad/Usabilidad_2010.pdf?attachauth=ANoY7cpboTgegk0jFpmofkXSt0r5XS_66GlxE DzDbzPAPkgh38POZn_tRc6so7O_rNv0yudprXARvyGzYCAGJEYBWqPabc BjpOC_kj_ugNBbINX6xFgObc8YZfu0niPwSPZVPveurgX7-rSkqGdQBoXAHBjpt1Y5rpMj0tb5xfZ_IGW3Bq0AUEbMDZVzF7zX_Ls5QBc wKZkSRAUjXg8RVzONq3X- jf97kzg9tllzj7Ql2afr04NwW_ZJ6ABVPhwrM5eIH-KDwlOR&attredirects=0)

[13] Ventajas y Desventajas de MYSQL | Super Info. Disponible en: <http://superinformacionweb.blogspot.com/2014/11/ventajas-y-desventajas-de-mysql.html>

[14] Perichart O., “Conteo de HCO” en Manual de lineamientos para la práctica de la nutrición clínica en enfermedades crónico degenerativas, Ed. Mc Graw Hill, México: 2012, pp. 165-166.