



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM ATLACOMULCO



“Aplicación móvil para apoyo al seguimiento y detección de riesgos en pacientes
con insuficiencia renal crónica”

T E S I S

Que para obtener el Título de:

Ingeniera en Computación

Presenta:

Aletvia Jacqueline León Venegas

Director de Tesis:

Dr. Everardo Efrén Granda Gutiérrez

Co-director de Tesis:

M. C. Juan Fernando García Mejía

Atlacomulco, México; Octubre del 2019

DEDICATORIAS

A mi padre por su apoyo, consejos, comprensión, amor y ayuda en los momentos más difíciles, por nunca dejarme sola y apoyarme en cada una de mis decisiones, por hacerme la persona que soy con valores, principios, carácter, perseverancia y coraje para abrirme camino a cada una de mis metas.

A mi madre, por ser siempre mi motivo para seguir adelante, porque pensar en ella siempre me dio las fuerzas necesarias para nunca rendirme, por ser mi mejor guía.

AGRADECIMIENTOS

A mi padre:

Por siempre creer en mí y por apoyarme a luchar por lo que quiero, por hacer de mí una mejor persona.

A mis asesores:

Por el tiempo, la dedicación y la paciencia empleados en la realización de este proyecto.

RESUMEN

Debido al aumento considerable de enfermedades renales, que además se presentan cada vez a una edad más temprana, se ha determinado como una herramienta de apoyo para el diagnóstico, seguimiento y fuente de información para los pacientes, el desarrollo de una aplicación móvil para el entorno del Sistema Operativo Android, que contiene como elemento fundamental un sistema experto basado en lógica difusa. Esta herramienta puede ayudar a los médicos en el diagnóstico de dichas enfermedades, al tiempo que provee información de orientación general para los pacientes.

En este proyecto, mediante el uso de la metodología de desarrollo de software en V y el Diseño Orientado a Objetos, se creó una aplicación móvil que permite a los usuarios conocer el porcentaje de riesgo que tienen de padecer una enfermedad renal. El proceso se realiza mediante una serie de preguntas sobre la actividad cotidiana y antecedentes familiares, que serán evaluadas por el sistema experto, para posteriormente generar una respuesta lo más acercada posible al diagnóstico emitido por un médico. Adicionalmente, permite guiar al paciente durante el seguimiento de su enfermedad, con información sobre algunas de las medidas de salud más importantes para su cuidado, así como le provee información sobre las enfermedades renales más comunes. En la fase de pruebas del sistema, se tomaron las respuestas de diferentes pacientes, observándose que el sistema funciona correctamente.

El diagnóstico de enfermedades renales es complicado para los médicos, ya que es afectada por múltiples factores. Debido a esto, muchas veces se dan tratamientos equivocados o incluso no se otorga un tratamiento oportuno, debido a la incertidumbre en su valoración. Por ello, un sistema experto es una forma eficiente de combatir dicho problema, en tanto que proporciona información cuantificable y de fácil acceso para los pacientes donde, si bien no se suple la experiencia del experto de la salud, si puede apoyar en la educación del paciente, como una herramienta útil para el cuidado de su padecimiento, así como un diagnóstico oportuno de un posible padecimiento renal.

Palabras clave: aplicación móvil, sistema de inferencia difuso, padecimientos renales.

ABSTRACT

Due to the noticeable increase in the occurrence of renal diseases, as well as their appearance in early stages of life, the development of an Android's environment mobile application for the diagnosis, disease monitoring and information for the patients is proposed in this document. The core component of this software application is an expert system based in fuzzy logic. This tool could help to physicians in the diagnosis of renal diseases and, at the same time, could provide a broad orientation to the patients.

In this project, a mobile application was created by using the V-model software development methodology as well as the Object-Oriented Design method. Such application allows to the users to estimate the risk probability of suffering a renal disease. The process is realized through a set of questions about the daily activity and family background, which are evaluated by the expert system; then, an estimated response is generated in a similar way to the physician diagnosis. Also, this application guides to the patient in the disease monitoring providing information about relevant health tips to help in the self-care and presents summarized information about the most common renal diseases. The correct functionality of the system was proved during the test phase by analyzing the responses of known patients.

The diagnosis of renal diseases is a complicated task to physicians because they are affected by multiple factors. Then, is common that the patient receives incorrect treatment or even the timely treatment is often not obtained due to the uncertainty of the diagnosis. In these cases, the use of an expert system could be an efficient way of fight against this problem, because it could provide quantifiable and ease-access information for the diagnosis. It is obvious that this application could not replace the experience of the health professional, but it could help in the education of the patient as a useful tool to the monitoring of the disease and the diagnosis of a renal disease.

Keywords: mobile application, fuzzy inference system, renal diseases.

ÍNDICE

DEDICATORIAS	i
AGRADECIMIENTOS	ii
RESUMEN.....	iii
ABSTRACT.....	iv
ÍNDICE	v
ÍNDICE DE TABLAS	vii
ÍNDICE DE FIGURAS.....	viii
1 INTRODUCCIÓN.....	1
2 PLANTEAMIENTO DEL PROBLEMA	3
2.1 Definición del problema.....	3
2.2 Objetivos de investigación	4
2.3 Preguntas de investigación	4
2.4 Justificación.....	4
2.5 Impactos	6
3 META DE INGENIERÍA.....	7
4 ESTADO DEL ARTE	8
4.1 Aplicaciones móviles en el área de la salud	8
4.2 Insuficiencia renal crónica.....	11
4.3 Algoritmos de inteligencia artificial	14
5 MÉTODO	26
5.1 Procedimiento de diseño e implementación de la aplicación.....	26
5.1.1 Modelo de desarrollo de software	27
5.1.2 Sistemas gestores de bases de datos para aplicaciones móviles	29
5.1.3 Diseño de la interfaz gráfica	31

5.1.4	Programación de la aplicación	40
5.2	Sistema de inferencia.....	41
5.2.1	Definición de conjuntos difusos.....	41
5.2.2	Función de inclusión	42
5.2.3	Inferencia difusa.....	45
5.3	Escala de Likert	47
6	RESULTADOS Y DISCUSIÓN	56
	CONCLUSIONES	68
	REFERENCIAS.....	70
	ANEXOS	75

ÍNDICE DE TABLAS

Tabla 1 Criterios RIFLE de definición y estratificación de riesgo en la disfunción renal aguda (Barrio, 2012).	20
Tabla 2 Criterios AKIN de definición y severidad de la disfunción renal aguda (Barrio, 2012).	21
Tabla 3. Matriz de referencias para determinar el estado del arte.	22
Tabla 4. Clases del paquete app	40
Tabla 5 Variables del modelo difuso	54
Tabla 6 Prueba al usuario 1	64
Tabla 7 Prueba al usuario 2	64
Tabla 8 Prueba al usuario 3	65

ÍNDICE DE FIGURAS

Figura 1 Algoritmo para el diagnóstico de insuficiencia renal aguda. (Anaya Fernández, et al., 2012).....	12
Figura 2 Algoritmo de diagnóstico y seguimiento de la enfermedad renal crónica (Anaya Fernandez, et al., 2012).....	15
Figura 3 Arquitectura de los MLPs (Delsol, et al., 2015).	16
Figura 4 Estructura de un individuo dentro de la población de un Algoritmo Genético (Delsol, et al., 2015).....	18
Figura 5 Mutación del individuo mediante MLP (Delsol, et al., 2015).....	19
Figura 6 Diagrama de flujo del proyecto	27
Figura 7 Modelo en V para el desarrollo de software	28
Figura 8 Mapa de navegación para la aplicación móvil.....	31
Figura 9 Información general sobre el funcionamiento de la aplicación.	32
Figura 11 Funcionamiento del apartado de diagnóstico.	32
Figura 10 Funcionamiento del apartado de seguimiento.	32
Figura 14 Registro del usuario.	33
Figura 12 Términos y condiciones que tendrá el uso de la aplicación.....	33
Figura 13 Inicio de sesión.	33
Figura 15 Menú principal.....	34
Figura 18 Comparación del órgano.....	35
Figura 17 Información de la enfermedad.	35
Figura 16 Clasificación de enfermedades renales.	35
Figura 20 Preguntas generadas para el diagnóstico.	36
Figura 21 Determinación del porcentaje de riesgo.	36
Figura 19 Información sobre el tratamiento a seguir.	36
Figura 23 Ajustes de la aplicación.	37
Figura 22 Perfil del usuario.....	37
Figura 24 Diagrama de casos de uso.....	38
Figura 25 Diagrama de clases de la aplicación móvil.....	39
Figura 26 Variable lingüística y restricciones difusas asociadas (Sáa, 2012).	42
Figura 27 Función de pertenencia de tipo trapezoidal.	43

Figura 28 Función de pertenencia de tipo triangular.	43
Figura 29 Función de pertenencia de tipo GAMMA.	44
Figura 30 Función de pertenencia de tipo L.....	44
Figura 31 Función de pertenencia de tipo S.....	44
Figura 32 Función de pertenencia de tipo Π	45
Figura 33 Conversión de la escala de Likert a Fuzzy (Sáa, 2012).....	48
Figura 34 Conjunto difuso para el número de veces que orina el usuario en 24 horas....	50
Figura 35 Conjunto difuso para el número de micciones por la noche.....	50
Figura 36 Conjunto difuso para el nivel de presión arterial.....	51
Figura 37 Introducción a la aplicación.....	56
Figura 38 Pantalla de inicio de sesión.....	57
Figura 39 Pantalla de registro de usuario.....	58
Figura 40 Menú de la aplicación.....	59
Figura 41 Pantalla de seguimiento del paciente.....	60
Figura 42 Pantalla de información.....	61
Figura 43 Agendar cita.....	62
Figura 44 Pantalla para el diagnóstico.....	63

1 INTRODUCCIÓN

El estilo de vida de los mexicanos ha generado una serie de problemáticas que desencadenaron el padecimiento de enfermedades como hipertensión, diabetes, enfermedades respiratorias, cardíacas, cáncer, insuficiencia renal, entre otras (de Paula Cerqueira, et al., 2014). Por ello se han desarrollado aplicaciones para el apoyo a la asistencia médica, por ejemplo, una aplicación móvil que permite conocer como la Web 2.0 está afectando la salud, guías de práctica clínica y recomendaciones, una aplicación que ayuda al diagnóstico y tratamiento de pacientes de todas las especialidades, consulta de fármacos, prescripción y todo lo relacionado a ellos, todas las antes mencionadas son dirigidas a profesionales médicos, las siguientes son orientadas a los pacientes, aplicaciones que permiten medir el ritmo cardíaco, pruebas de audiometría, guías de primeros auxilios, recordatorios de medicamentos, algunas permiten buscar profesionales y centros médicos, consulta de medicamentos mediante código de barras, monitorización de actividad diaria, control de hábitos alimenticios, entrenamiento cerebral, clasificación de tumores cancerígenos y clasificación de diabetes de tipo II; sin embargo, existen muchas enfermedades que pueden ser apoyadas por este tipo de tecnología, por lo que la necesidad de desarrollar aplicaciones de ayuda sigue siendo vigente.

Existe un alto porcentaje de individuos que padecen alguna enfermedad renal, debido a los malos hábitos alimenticios, a la falta de actividad física y a los antecedentes familiares que presenta el paciente (Vergottini, et al., 2013). Por tal motivo se pretende generar una aplicación orientada a las enfermedades renales que, actualmente, las enfermedades renales son una de las que afectan en mayor medida a la población (CNN, 2015).

Anteriormente la calidad de vida de las personas con este tipo de enfermedades se consideraba mala, pero con los aportes de la tecnología en este sector, se ha logrado mejorar, lo cual representa mayores expectativas de vida para los pacientes y más manejabilidad de la enfermedad. No obstante, es necesario continuar con los avances para lograr satisfacer las necesidades de personas con otras enfermedades que aún están pendientes, por ello se pretende enfocar el trabajo en las enfermedades renales ya que es un campo en el que no todas las necesidades han sido satisfechas por la tecnología.

Debido a la falta de documentación en la población, es de vital importancia que todos tengan un fácil acceso a información confiable y de calidad sobre una enfermedad, por ello se deben desarrollar aplicaciones que cumplan con los métodos de evaluación para que de esta forma se tenga la certeza de que los usuarios podrán tener privacidad con los datos que proporcionan a la aplicación y que los resultados que obtengan de la misma podrán ser de ayuda para cuando se acuda a un médico a hacer el diagnóstico correspondiente de acuerdo a la sintomatología y antecedentes que presente el paciente.

Mediante esta aplicación, es posible que las personas tengan mayor conocimiento sobre las enfermedades renales, debido a que muchas veces son propensos a padecerlas y no tienen conocimiento de ello, esta herramienta desarrollada, dará un porcentaje que permita conocer el riesgo que se tiene, las pruebas realizadas a diversas personas permiten conocer el tiempo de respuesta de la aplicación, así como la veracidad de los resultados generados por el algoritmo.

Debido a que cada vez han aumentado más el número de personas con enfermedades renales, era necesario desarrollar una herramienta que permitiera simular el conocimiento humano en dicha área, por ello un sistema experto fue el más adecuado para dicha tarea, el proceso es subjetivo y permite optimizar el diagnóstico.

2 PLANTEAMIENTO DEL PROBLEMA

2.1 Definición del problema

A lo largo de los años se han desarrollado diversas técnicas de inteligencia artificial para la solución de problemas en el campo de la medicina; todas ellas con una finalidad en específico. Algunos ejemplos se encuentran en las siguientes áreas: a) cirugía, donde se pueden encontrar los robots que son capaces de hacer intervenciones con mayor precisión que un humano y brindando a los pacientes la oportunidad de recuperarse mucho más rápido; b) en terapéutica (EcuRed, 2019) (área que se encarga de aplicar los medicamentos u otros medios para el tratamiento de enfermedades.), se encuentran las prótesis que, mediante información sensorial, permite a las personas rehabilitarse en un periodo de tiempo considerablemente corto; c) en la detección de nuevas enfermedades, como el detectar cuando el ritmo cardiaco es inconsistente y puede representar un riesgo; d) también en el seguimiento de los pacientes, lo cual permite hacer una comparación entre el estado inicial del paciente y el avance que ha tenido después de ser sometido a algún tratamiento.

Gran parte de la población es propensa a padecer algún tipo de enfermedad renal, debido a los hábitos alimenticios y al ritmo de vida que llevan. Por lo que se busca desarrollar una alternativa de solución que potencialmente puede ayudar a la población, brindándole información, tratamientos, limitaciones y calidad de vida de estas personas, ayudando a crear conciencia sobre el cuidado de la salud. Cabe mencionar que el enfoque de este proyecto se encuentra en el desarrollo de la aplicación, dejando como trabajo a futuro el análisis de su impacto en la población

Desde el punto de vista computacional, el uso de herramientas tecnológicas asociadas al desarrollo de aplicaciones para los teléfonos inteligentes, así como de herramientas de soft computing, son elementos que tienen potencial para propiciar la incorporación de metodologías basadas en la forma de pensamiento humano que buscan dar soluciones a ciertos problemas, esto mediante la interacción eficiente con sistemas complejos cuando la información con la que se cuenta no es suficiente.

2.2 Objetivos de investigación

Objetivo general:

Desarrollar un sistema experto, basado en lógica difusa, que funcione como auxiliar en el seguimiento y detección de riesgos en pacientes con enfermedades renales, mediante una aplicación intuitiva diseñada para dispositivos móviles con sistema operativo Android.

Objetivos específicos:

1. Determinar los parámetros de riesgo de las enfermedades renales, que se considerarán para el desarrollo de esta aplicación, mediante la asesoría de un médico.
2. Identificar los requerimientos de una aplicación para el seguimiento de pacientes con problemas renales.
3. Codificar un algoritmo de softcomputing, mediante lógica difusa, desarrollado en Android Studio para diagnóstico de enfermedades renales.
4. Validar el funcionamiento de la aplicación de acuerdo con los requerimientos establecidos en el objetivo 1.

2.3 Preguntas de investigación

- ¿Cómo se pueden diagnosticar oportunamente las enfermedades renales?
- ¿Qué variables se utilizan para determinar el riesgo de padecer una enfermedad renal?
- ¿Qué técnica de softcomputing se utilizaría como herramienta tecnológica para el diagnóstico de este tipo de enfermedades?

2.4 Justificación

Según el Estudio de la Insuficiencia Renal en México, al menos 40,000 nuevos casos de insuficiencia renal se suman cada año en el país, esto debido a la falta de cultura de

prevención. De acuerdo con este estudio, México es el único país en el que se registran niveles tan altos con 500 enfermos por cada millón de habitantes (Clínica Renalis, 2016).

Durante los últimos años el número de personas que padecen insuficiencia renal ha aumentado, representando un 17% de la población mayor a 20 años (Vergottini, et al., 2013) y es necesario prevenir a la población sobre esta enfermedad, documentarlos y de esta manera ayudar a la población a cambiar sus hábitos alimenticios y las rutinas diarias, todo esto con la finalidad de evitar que los individuos sean propensos a padecer ésta y muchas otras enfermedades.

El estilo de vida en México ha provocado que muchas personas padezcan enfermedades como obesidad, diabetes, hipertensión, etc. que son principios de muchas otras enfermedades. Es necesario conocer los antecedentes familiares ya que pueden representar un factor de riesgo para las personas y requieran asistir al médico correspondiente.

Actualmente existen varias aplicaciones similares a ésta, sin embargo, están enfocadas a otro tipo de enfermedades, por ello surgió la idea de implementar esta aplicación, la cual se espera que sea de gran ayuda para las personas que en algún momento decidan utilizarla, también se pretende facilitar el análisis del tratamiento, progreso, etc. Así los médicos correspondientes podrán diagnosticar la enfermedad a tiempo y potencialmente se puede apoyar en la disminución del riesgo de que las personas asistan a revisión cuando ya se haya desarrollado el problema.

El desarrollo de aplicaciones en el área de salud ha representado un avance computacional importante debido a que facilitan el trabajo del personal médico ya sea para generar un diagnóstico, llevar a cabo un seguimiento de tratamiento, consulta de información, entre otras. Por ello la ingeniería es un campo de gran importancia para el desarrollo de ésta y otras áreas de aplicación.

Existen aplicaciones que realizan un análisis del seguimiento de padecimientos como diabetes, hipertensión, enfermedades respiratorias, cardíacas, entre muchas otras, sin embargo, actualmente no existe una aplicación orientada a las aplicaciones renales y debido al alto índice de personas que sufren este padecimiento, se considera de gran importancia que la población tenga más información sobre los riesgos, calidad de vida y

tratamientos para que estén más familiarizados con el tema, ya que no todos los individuos tienen la capacidad de acceder a información confiable sobre este aspecto.

En este proyecto se utilizará la programación en Android para el desarrollo de la aplicación, debido a que el 85.9% de las personas en el mundo cuentan con un dispositivo móvil y tiene este sistema operativo (Pascual, 2018).

Con la creación de esta aplicación se pretende beneficiar a toda la población que decida hacer uso de ella, ya que les permitirá conocer el nivel de riesgo que tienen de padecer alguna enfermedad renal, obtener información de todo lo relacionado con ella, llevar el control si es que ya tienen el padecimiento, entre otras funciones que permitirán al usuario tener mayor control de su salud.

2.5 Impactos

- Científico: Se pretenden utilizar técnicas de softcomputing para mejorar la metodología utilizada para la detección de riesgos de padecer una enfermedad renal, lo cual representa que la información esté al alcance de todas las personas que cuenten con un dispositivo móvil y sea confiable.
- Tecnológico: Se diseñará una aplicación móvil desarrollada en Android que permitirá a la población tener información acerca de las enfermedades renales y detectar los riesgos a tiempo para tomar el tratamiento adecuado y así se verá beneficiada la calidad de vida de los pacientes.
- Económico: Aunque se considera únicamente como impacto potencial se prevé que se reducirá el costo de salud, debido a que auxiliará a las personas a llevar un mejor control de su enfermedad, por eso se desea llevar la información necesaria a todas partes para que las personas estén informadas, de esta forma mejoren sus hábitos y reduzcan los riesgos de padecer esta y otras enfermedades (Plan Seguro, 2019).
- Social: Se desea lograr que esta aplicación sea manejable, sencilla, útil y atractiva para todas aquellas personas que la utilicen sin importar su edad o sexo. Principalmente se pretende lograr que las personas tengan un mayor control de los riesgos, seguimiento y tratamiento de las enfermedades renales y de esta forma mejorar su calidad de vida.

3 META DE INGENIERÍA

El desarrollo de un aplicación móvil para el apoyo a la detección de riesgos a desarrollar una enfermedad renal estará diseñado con dos módulos principales, el primero de ellos estará codificado con un algoritmo que, mediante el análisis de los antecedentes familiares, hábitos diarios y alimenticios, permita que la población actual propensa a presentar este padecimiento pueda tener un indicio del porcentaje de amenaza con el que cuenta y de esta forma acuda a una valoración médica para hacer la detección temprana de la enfermedad, se diseñará una interfaz amigable con el usuario que mediante preguntas, haga la evaluación y pueda de esta forma determinar el nivel de riesgo que presenta el usuario.

El segundo de ellos permitirá llevar el control de la enfermedad una vez diagnosticada, se podrá ingresar el nombre y dosis de los medicamentos que son administrados para el control del padecimiento, se podrá hacer el registro de estudios y próximas citas que se tengan para que los usuarios tengan el recordatorio y lleven un seguimiento constante.

La implementación de este software pretende hacer que las personas que utilicen la aplicación tengan un indicio del riesgo que tienen de padecer insuficiencia renal en cualquiera de sus etapas, esto con la finalidad de que las personas puedan realizarse una valoración médica para verificar el diagnóstico y de esta forma puedan atender el padecimiento a tiempo.

La aplicación no pretende sustituir el diagnóstico de un médico experto, únicamente será desarrollada para apoyo al diagnóstico y de esta forma permitir a los usuarios que asistan al médico para estar en constante revisión para evitar cualquier tipo de complicación en la salud. Se debe tener siempre presente que, sin importar el resultado de la valoración del software, únicamente es un indicio, lo que hace referencia a que debe ser constatado con un médico experto de la salud.

4 ESTADO DEL ARTE

4.1 Aplicaciones móviles en el área de la salud

La insuficiencia renal ha sido una enfermedad que, al igual que el cáncer, ha causado muchos fallecimientos; sin embargo, gracias a los estudios realizados y a los avances tecnológicos, se ha podido reducir estos decesos, dando mayores oportunidades de mejora a las personas que padecen esta enfermedad.

Actualmente la calidad de vida de los pacientes con enfermedades renales ha mejorado considerable gracias a los tratamientos, herramientas tecnológicas que permiten realizar funciones que su organismo ya no hace debido a la atrofia de los riñones, entre otras. Sin embargo, no siempre se tienen las mejores condiciones para sobrellevar una enfermedad de este tipo, no obstante, el uso de una aplicación orientada al sector salud no pretende sustituir a los médicos en su labor, sino que es un auxiliar en la complementación de un diagnóstico médico, tratamiento y seguimiento de una enfermedad (Morais Arruda Costa, et al., 2016).

El área de la medicina se ha visto beneficiada por la tecnología; esto ha permitido tener un gran avance en el hallazgo de nuevas enfermedades, en la implementación de robots para fines quirúrgicos prótesis para la rehabilitación de personas con lesiones graves, entre otras muchas aplicaciones, todas ellas han permitido que las personas que estén en contacto con estas tecnologías mejoren considerablemente su calidad de vida con una inversión económica relativamente menor que la que se requería en otros tiempos.

Hoy en día se han desarrollado aplicaciones móviles enfocadas al área de salud, las cuales cuentan con características, entre las más comunes, permiten llevar un seguimiento de enfermedades específicas como por ejemplo cáncer de mama, algunas otras permiten analizar la frecuencia cardiaca de una persona y determinar si está en riesgo de sufrir un infarto (Alonso Arévalo, 2016), estas son algunas de las aplicaciones que están actualmente en el mercado, sin embargo, no se han desarrollado aplicaciones que permitan detectar los riesgos de padecer una enfermedad renal, por lo cual es necesario prestar atención a este sector de la población ya que esta es una de las enfermedades más comunes en nuestros días.

Todos los avances han tenido como objetivo en común mejorar las condiciones, acelerar los procesos y tener mayor precisión a la hora de realizar una intervención o presentar un diagnóstico. El desarrollo de aplicaciones móviles en el campo de la salud permite tener la información mucho más accesible de lo que era antes, únicamente se necesita promover entre la población el uso de estas para que pueda sacar el mayor provecho posible a cada una de las aplicaciones que están al alcance.

Para la creación de aplicaciones móviles orientadas a la medicina es conveniente cumplir con diversos requisitos, como son (Velasco Rodríguez, 2018);

1. Las agencias de calidad Sanitaria otorgan los distintivos *AppSaludable*, la cual mide usabilidad, calidad, seguridad, calidad, servicios y privacidad.
2. El App Date, es un sitio que permite que profesionistas de diversas áreas generen un debate sobre las tendencias en la creación de aplicaciones que ayuden a solucionar problemas específicos de los usuarios, realiza un trabajo de investigación y evaluación con el objetivo de informarse acerca de las aplicaciones móviles en el marco de la *mHealth*, término que se le da a la práctica de medicina y a la salud con el apoyo de cualquier dispositivo móvil.
3. Índice *iSYScore* que tiene por objetivo recomendar una aplicación, clasificar la puntuación y dar criterios para el continuo mejoramiento de estas.

De los puntos anteriores, cabe destacar al índice *iSYScore*, que es un instrumento de medida específicamente desarrollado para evaluar las aplicaciones de salud, las cuales reciben una ponderación por expertos en comunicación, salud e internet. El *iSYScore* plantea indicadores de tres dimensiones (Grau, et al., 2016):

1. Interés popular (Se puede colocar una puntuación máxima de hasta 11 puntos): Se evalúa que las aplicaciones tengan una confección multiplataforma, es decir, que estén disponibles en los dos sistemas operativos, iOS y Android. La puntuación se desglosa de la siguiente manera:
 - a. Los usuarios evalúan positivamente la aplicación: 4 puntos.
 - b. Está disponible en dos plataformas: 3 puntos.
 - c. Declarado de interés por alguna dependencia: 4 puntos.

2. Confianza (puntuación máxima de hasta 18 puntos); son aquellas aplicaciones don de los indicadores apuntan a un contenido de calidad. En este apartado se toman en cuenta los sellos de calidad que estén reconocidos, en caso de haberlos, la aplicación obtiene directamente los puntos máximos, en caso contrario la puntuación se dividirá en los siguientes ítems:
 - a. Validación por un profesional especializado, organismo sanitario o sociedad científica: 4 puntos.
 - b. Promovido por una asociación de personas que son usuarios: 3 puntos.
 - c. La aplicación cuenta con un *website* y con el compromiso de la protección de datos: 4 puntos.
 - d. Cuenta con la citación de fuentes de evidencia: 4 puntos.
 - e. Se hace referencia a la organización responsable: 3 puntos.
3. Utilidad (puntuación máxima de hasta 18 puntos): La puntuación máxima deriva de una investigación debidamente sustentada, la cual demuestre que la aplicación es útil. Sin embargo, esta condición es infrecuente, es por ello por lo que se tomaron en cuenta diversos parámetros derivados de la búsqueda de antecedentes sobre cuando deban mostrarse resultados:
 - a. Investigación sobre una muestra menor de 30 usuarios: 3 puntos.
 - b. Declaración de una sociedad científica o asociación de usuarios: 3 puntos.
 - c. Proporción de información: 3 puntos.
 - d. Proporciona seguimiento útil en salud (*trackers*): 3 puntos.
 - e. Enlaza con otros usuarios: 3 puntos.
 - f. Hace uso de juegos para promocionar la salud: 3 puntos.

Estos métodos son evaluados por personal experimentado en cada una de las áreas, dichas normas determinarán la confiabilidad de la aplicación, el rendimiento, la privacidad de los datos y que los datos que muestre al realizar algún análisis sean los correctos.

Por otro lado, debido a que el desarrollo de aplicaciones para la salud está destinadas a mantener y mejorar la calidad de vida y el bienestar de las personas, para estas aplicaciones móviles se ha implementado un término que hace referencia a estas tecnologías, *mHealth*. (Velasco Rodríguez, 2018).

4.2 Insuficiencia renal crónica

La insuficiencia renal crónica es una alteración en la estructura renal persistente por al menos tres meses o presencia de filtrados, los cuales se clasifican en cinco estadios (Anaya Fernandez, et al., 2012):

1. Lesión renal con filtrado glomerular o aumentado.
2. Lesión renal con descenso leve del filtrado glomerular.
3. Descenso moderado del filtrado glomerular.
4. Descenso importante del filtrado glomerular.
5. Insuficiencia renal, con filtrado glomerular.

A diferencia de otras referencias, este artículo plantea que las causas más frecuentes de padecer una enfermedad renal son la nefropatía diabética y las nefropatías vasculares. A continuación, en la Figura 1 se muestra el algoritmo utilizado para el diagnóstico de la insuficiencia renal aguda.

El algoritmo tiene como principal objetivo determinar las diferencias entre los pacientes que padecen insuficiencia renal aguda y enfermedad renal crónica, una vez entendido esto, se procede a hacer el diagnóstico de la enfermedad, así como el seguimiento de esta.

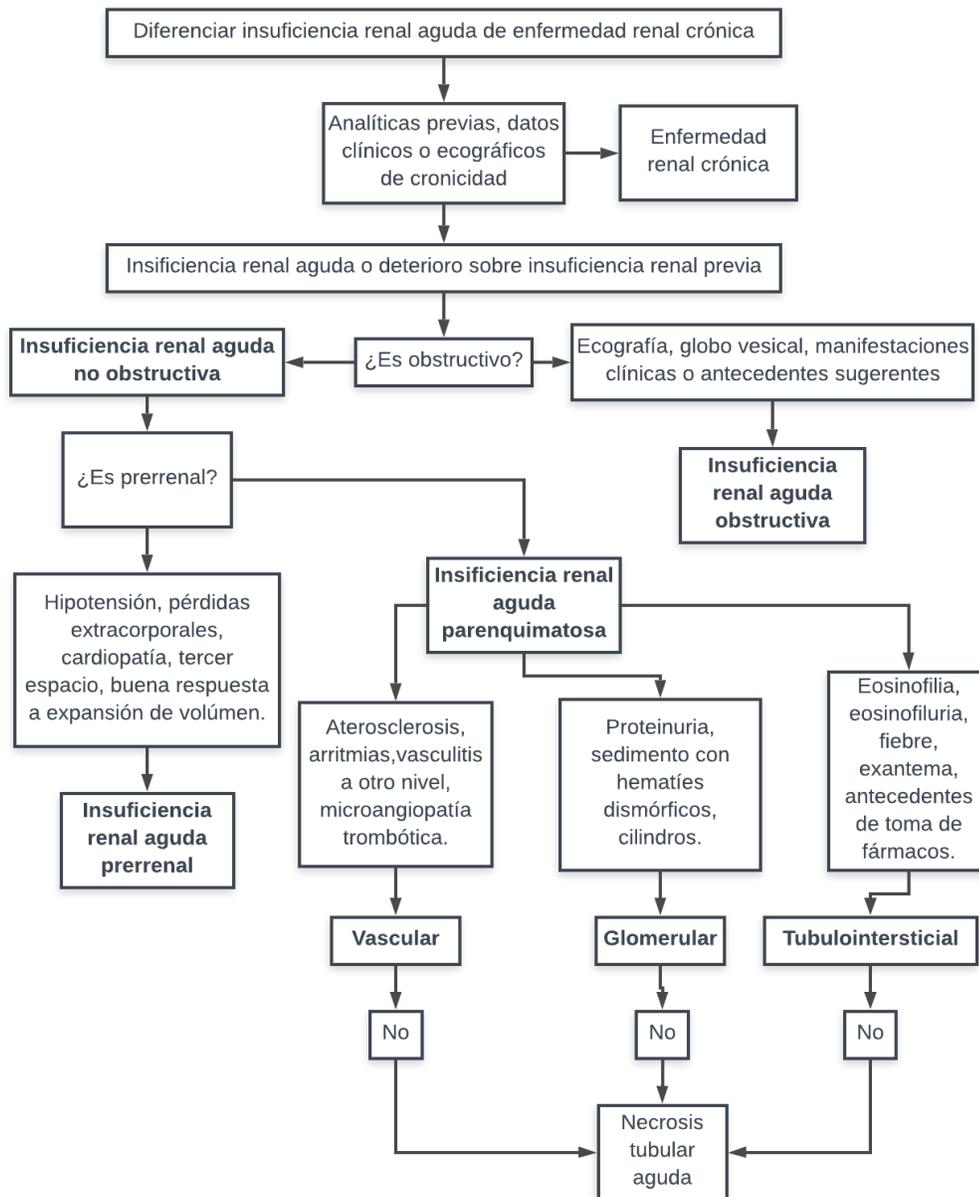


Figura 1 Algoritmo para el diagnóstico de insuficiencia renal aguda. (Anaya Fernández, et al., 2012).

Cualquier problemática renal se considera como causa potencial de padecer una enfermedad renal crónica; es por ello que se debe estar en contacto con un médico experto, para que realice las pruebas necesarias y pueda determinar si se tiene alguna enfermedad

y el tipo de tratamiento que se debe seguir para contrarrestar los efectos de la enfermedad, dando así mayor esperanza de vida para los pacientes (Martínez Castelo, et al., 2014).

En gran parte de la literatura se menciona que muchos de los factores que se toman en cuenta para el diagnóstico de las enfermedades renales son los antecedentes patológicos de los familiares; sin embargo, los autores mencionan en el algoritmo que únicamente se toman en cuenta los resultados obtenidos después de hacer una serie de estudios que podrán dar a conocer en qué etapa de la enfermedad se encuentran y determinar así el tratamiento adecuado para cada uno de ellos.

Según el (Instituto Mexicano del Seguro Social, 2019), motivar y apoyar a los pacientes permite mejorar en ellos los resultados de salud, la calidad de vida y la adaptación al tratamiento. Al igual que literaturas desarrolladas anteriormente, este artículo hace referencia a que los malos hábitos alimenticios y el estilo de vida son factores negativos para el desarrollo y avance del daño renal, por lo que se deben proporcionar asesorías nutricionales a todas las personas propensas a padecer una enfermedad renal, motivarlos a ejercitarse para mantener el peso adecuado, así como evitar el consumo de bebidas embriagantes y tabaco.

Para la evaluación de los pacientes con riesgo a padecer enfermedades renales se debe realizar:

1. Medición de la tensión arterial.
2. Medición de la creatinina sérica (CrS).
3. Estimación de la tasa de filtrado glomerular (TFG).
4. Evaluar la presencia de marcadores de daño renal.
 - a. Albuminuria-proteinuria.
 - b. Análisis del sedimento urinario.
 - c. Estudios de imagen o histopatológicos.

Se mencionan que la nefro-protección es una estrategia que permite incluir tratamiento farmacológico y no farmacológico para revertir el desarrollo del daño renal. Estas medidas, la mayor parte de las veces, hacen uso de antihipertensivos, nefrotóxicos, control de glucosa en caso de las personas que también padecen diabetes, control de peso, entre

otros. Todas estas medidas de protección tienen mucho más impacto cuando son monitoreados y atendidos en las primeras etapas del diagnóstico de la enfermedad.

Uno de los factores importantes en el desarrollo de la enfermedad es el uso excesivo de los medicamentos, ya que representan hasta el 20% de los casos de insuficiencia renal, ya sea aguda o crónica. Por ello, debe evitarse en todo momento el uso de drogas u otras sustancias tóxicas para personas con riesgos renales, a menos que estén debidamente indicados por médicos especialistas.

A continuación, se muestra en la Figura 2 el algoritmo que permite el diagnóstico y seguimiento de la enfermedad renal crónica.

4.3 Algoritmos de inteligencia artificial

En la literatura se ha identificado una aplicación para detectar si un tumor es benigno o maligno, mediante muestras tomadas en pacientes propensos a padecer cáncer de mama, al igual que para clasificar en negativo o positivo las muestras correspondientes de pacientes con riesgos de tener diabetes tipo II (Delsol, et al., 2015).

Para este propósito se hizo uso de perceptrones multicapa, algoritmos genéticos y algoritmos que hibridan ambas técnicas.

Las redes neuronales artificiales (RNAs), los algoritmos híbridos y las metaheurísticas, a lo largo del tiempo se han convertido en una herramienta efectiva e indispensable para la resolución de problemas, todo esto es gracias a la estructura con la que cuentan que son: paralelas y distribuidas. Las RNAs tienen la capacidad de aprender, generalizar y aproximar de manera no lineal, esto las hace ser adecuadas para resolver problemas de clasificación.

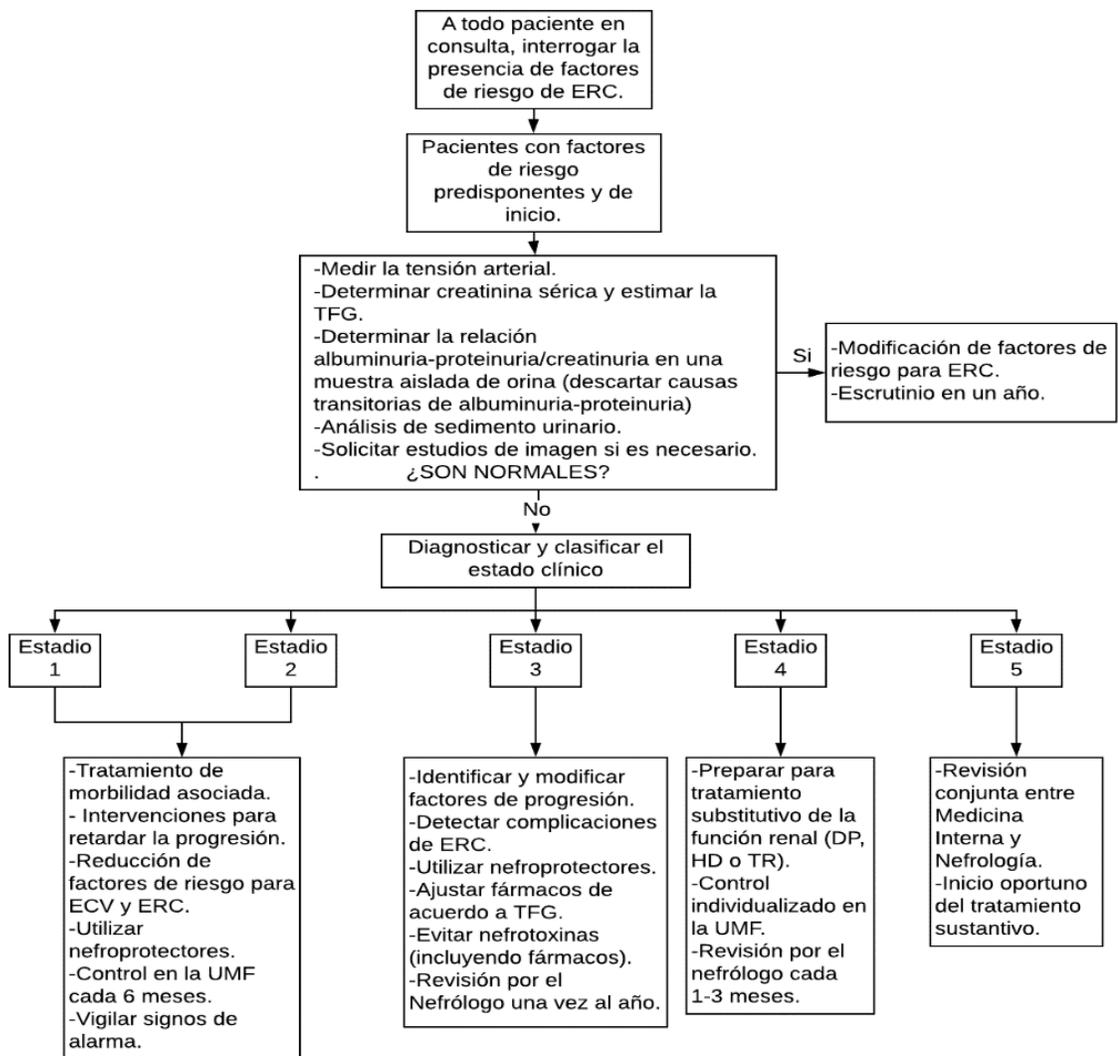


Figura 2 Algoritmo de diagnóstico y seguimiento de la enfermedad renal crónica (Anaya Fernandez, et al., 2012).

Una de las técnicas aplicadas en los problemas de clasificación es el perceptrón multicapa (*Multi-Layer Perceptron* o MLP), que es una red *feed-forward* (con realimentación positiva) con aprendizaje supervisado. Uno de los métodos de aprendizaje más utilizado es el método *backpropagation* o propagación del error hacia atrás; dicho método ha permitido ampliar las opciones de aplicación de los MLPs. Algunos ejemplos se pueden ver en el diagnóstico de enfermedades, clasificación de señales, simulación de la dinámica del brazo de un robot, puntuación para la solicitud de créditos, entre otros. El

funcionamiento de este tipo de aprendizaje está basado en la siguiente metodología (Delsol, et al., 2015):

1. Se aplica un patrón de entrada para estimular la primera capa de la red neuronal
2. La respuesta a este patrón se propaga a todas las capas ocultas de la red hasta generar una respuesta final; esta propagación se realiza al evaluar la suma ponderada en la función de activación. Dicha suma surge de la combinación lineal de las entradas y los pesos sinápticos correspondientes a cada una.
3. Se evalúa el error al comparar el resultado obtenido en las neuronas de salida con el patrón, mediante una función de costo.
4. Los errores obtenidos se transmiten hacia atrás a cada una de las neuronas de las capas, y se reajustan los pesos de conexión de cada neurona; esto representará que se acerque más a la salida deseada con cada repetición o *momentum*, el cual tiene la capacidad de acelerar la convergencia del algoritmo.

En la Figura 3 se representa la arquitectura de los MLPs donde In_n representa las neuronas de entrada, m las neuronas de la capa oculta, w son los pesos sinápticos, representados por dos matrices: w_{ij}^1 (con $i = 1, \dots, m$ y $j = 1, \dots, n$) y w_{1j}^2 (con $j = 1, \dots, m$) y Out representa la salida de la red neuronal.

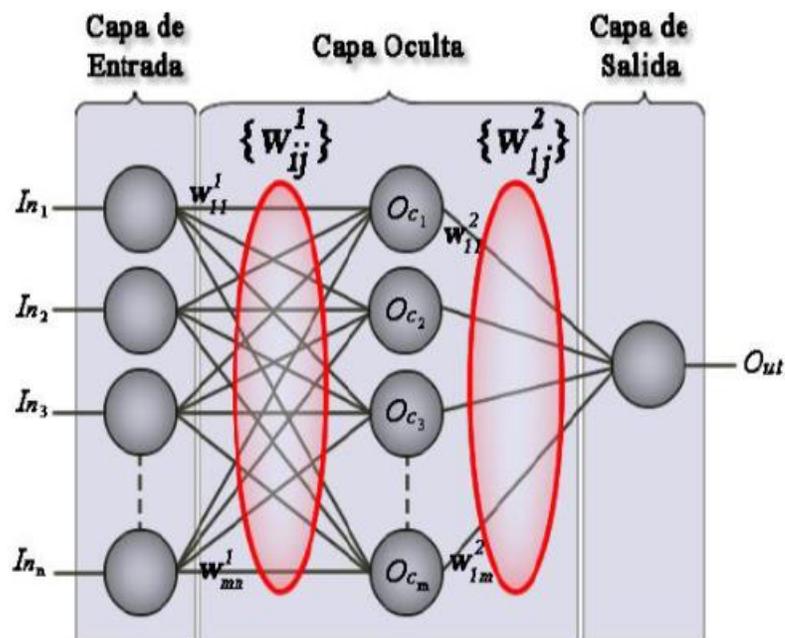


Figura 3 Arquitectura de los MLPs (Delsol, et al., 2015).

Otra de las técnicas aplicadas al diagnóstico de tumores de mama son los algoritmos genéticos (*Genetic Algorithms* o *GAs*), que tienen como propósito la simulación de la evolución de los individuos mediante la selección natural Darwiniana (Delsol, et al., 2015). En términos computacionales, hace referencia a la realización de una búsqueda multidireccional, pero siempre manteniendo una población de posibles soluciones, formando información mediante la mutación e intercambiándola en el cruzamiento de las soluciones.

El cruzamiento de los individuos se realiza a través de diferentes operadores probabilísticos, por ejemplo: ruleta, ranking, torneo, entre otros. Cada vez que el algoritmo es ejecutado, los individuos menos aptos de la población actual son sustituidos por las mejores creaciones, para mantener a las mejores opciones e ir eliminando a los que no representan una solución óptima; a este proceso se le conoce como elitismo.

Existe una función *fitness* que permite medir el promedio de error al momento de la clasificación. La evaluación del *fitness* es equivalente al cálculo del error medio de una MLP durante una ejecución.

En la Figura 4 se observa la estructura de los genes de cada individuo, cada uno de ellos hace referencia a los pesos sinápticos de un MLP, en el caso de los algoritmos genéticos se estructura en forma de vector a diferencia de los MLP que tienen una estructura matricial. Cada peso de conexión de las MLP presenta un gen del individuo, el cual es creado con los valores aleatorios y teniendo una distribución uniforme.

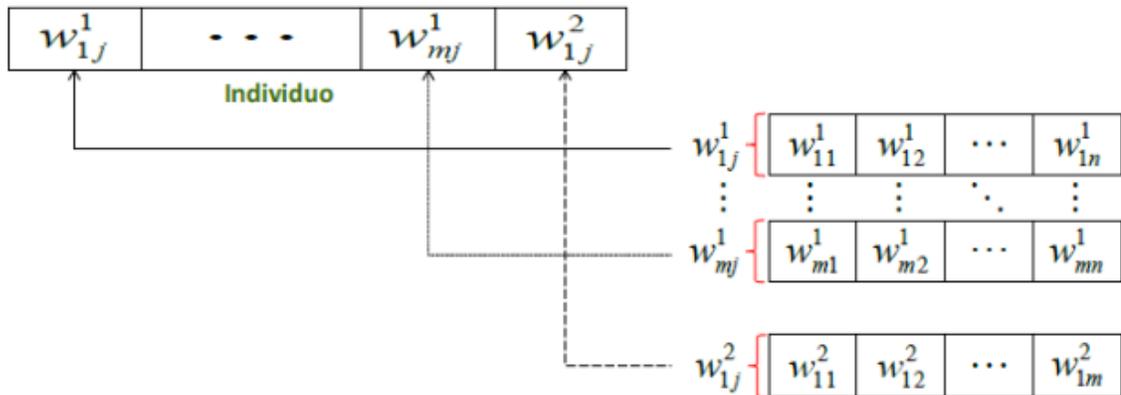


Figura 4 Estructura de un individuo dentro de la población de un Algoritmo Genético (Delsol, et al., 2015).

Finalmente, las metaheurísticas híbridas representan una combinación de algoritmos metaheurísticos, programación matemática y aprendizaje automático o *machine learning*; estas técnicas permiten la realización de búsquedas exhaustivas para encontrar la mejor solución a un problema.

Para el caso de la detección y clasificación de cáncer de mama y diabetes, se han implementado las técnicas antes mencionadas, cada una realizando tareas específicas, pero trabajando juntas para un fin en común. Los algoritmos genéticos proporcionan la estructura de contención, mientras que el MLP será el operador de mutación, lo cual consiste en hacer una mutación inteligente que intentará minimizar el error de clasificación, si esto se cumple, el individuo aumentará la probabilidad de sobrevivir durante la ejecución del algoritmo.

En la Figura 5 se observa el proceso de mutación, que se realiza ingresando un individuo en la red y entrenándola para que el *fitness* mejore.

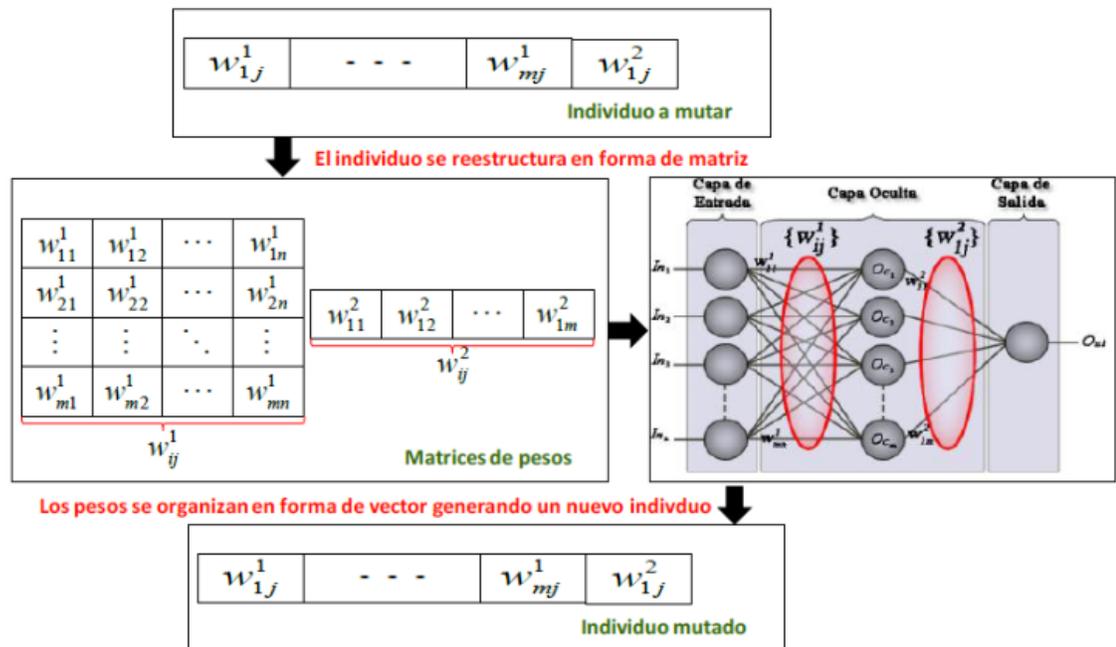


Figura 5 Mutación del individuo mediante MLP (Delsol, et al., 2015).

Otra aplicación de inteligencia artificial es al analizar la funcionalidad renal de las personas con cirrosis. La reducción de orina en pacientes con esta problemática representa problemas serios, por ello, la utilización de cambios dinámicos en la creatinina resulta ser un punto clave en el diagnóstico de lesiones renales agudas. Por tal motivo se realizaron mejoras al algoritmo para la definición dinámica de la insuficiencia renal aguda en pacientes con cirrosis, y hoy en día se realizaron algunas innovaciones que permiten hacer el algoritmo mucho más eficiente, algunas de ellas se enlistan a continuación (Fundación clínica para la reseva biomédica, 2015):

1. Utilización de los cambios dinámicos en los valores de la creatinina en suero proveniente de los criterios *KDIGO* (*Kidney Disease: Improving Global Outcomes*).
2. Permite una aplicación racional de los recursos terapéuticos, evitando posibles consecuencias de sobretreatmento de las lesiones renales agudas.
3. Eliminación de valores de corte de la creatinina en suero para el diagnóstico del síndrome hepatorenal en los criterios *AKIN* (*Acute Kidney Injury Network*).

Las guías internacionales del consorcio KDIGO tienen como objetivo guiar el diagnóstico, la evaluación, el manejo y el tratamiento de pacientes con enfermedades renales crónicas, también se hacen recomendaciones para el manejo de pacientes y los criterios de derivación (Gorostidi, et al., 2014).

El grupo AKIN propuso una evaluación de los criterios para el diagnóstico y clasificación de severidad basados en los criterios RIFLE (Risk, Injury, Failure, Loss y End stage kidney disease) para los problemas de insuficiencia renal aguda. Algunos de los criterios incluyen el perfil temporal, el incremento de CrS y la disminución del volumen de diuresis y se añade el incremento absoluto de CrS.

Los criterios RIFLE consisten en tres niveles de disfunción (riesgo, daño y fallo) de acuerdo con el incremento en CrS o con la disminución del filtrado glomerular (FGe), así como con la disminución del volumen de diuresis y en dos medidas de resultado (pérdida y enfermedad renal terminal) en función al tiempo de dependencia del tratamiento (Barrio, 2012). Los criterios se especifican en la Tabla 1.

Tabla 1 Criterios RIFLE de definición y estratificación de riesgo en la disfunción renal aguda (Barrio, 2012).

	Criterios de FGe	Criterio de volumen de diuresis
Riesgo	Incremento de CrS > 1.5 sobre basal y/o reducción de FGe 25%	< 0.5ml/kg/h en 6 horas
Daño	Incremento de CrS > 2 sobre basal y/o reducción de FGe 50%	< 0.5ml/kg/h en 12 horas
Fallo	Incremento de CrS > 3 sobre basal y/o reducción de FGe 75%	< 0.3ml/kg/h en 24 horas o anuria 12 horas
Pérdida	Necesidad de técnicas de depuración extrarrenal (TDE) > 4 semanas	
Enfermedad renal terminal	Necesidad de TDE > 3 meses	

Debido a que los criterios iniciales presentaban diversas limitaciones, el grupo AKIN propuso una revisión de los criterios y la modificación de estos para la insuficiencia renal aguda. La modificación de los criterios incluye un perfil temporal menor a 48 horas, supervisión del aumento en un 50% de CrS, en la disminución de 0.5ml/h en un periodo de 6 horas del volumen de diuresis y en el incremento absoluto mayor a 0.3mg/dl de CrS, esto debido a que se realizaron investigaciones en las cuales se demostró que el incremento mínimo en la cifra de CrS es un predictor de mortalidad y estancia médica hospitalaria. Los criterios se especifican en la Tabla 2.

Tabla 2 Criterios AKIN de definición y severidad de la disfunción renal aguda (Barrio, 2012).

Estadio	Criterio CrS	Criterio de volumen de diuresis
1	Incremento de CrS \geq 0.3mg/dl (26.4mcmol/L) o aumento de 1.5 a 2 veces sobre basal	$<$ 0.5ml/kg/h en 6 horas
2	Incremento de CrS de 2 a 3 veces sobre basal	$<$ 0.5ml/kg/h en 12 horas
3	Incremento de CrS $>$ 3 veces sobre basal o CrS \geq 4.0mg/dl ($>$ 354mcmol/L) con un aumento de al menos 0.5 mg/dl (44 mcmol/L)	$<$ 0.3ml/kg/h en 24 horas o anuria 12 horas

A continuación, se muestra, en la Tabla 3, la matriz de referencias, en la cual se presenta la bibliografía que fue utilizada para la discusión del estado del arte, en ella se determinan las ideas principales del contenido que se desarrolla en cada una de las citas.

Tabla 3. Matriz de referencias para determinar el estado del arte.

Fuente	Resumen	Metodología	Áreas de oportunidad
Vergottini, J. C. y otros, 2013. Detección de riesgo para enfermedad renal en adolescentes. <i>Revista de Salud Pública</i> , julio, Volumen 17, pp. 63-72.	El índice a nivel mundial de la población adulta que padece insuficiencia renal crónica ha ido aumentando a lo largo del tiempo, por tal motivo es necesario alertar sobre los riesgos de no tener hábitos saludables, así como brindarles información para identificar los síntomas que pueden presentarse.	Se realizó un estudio con alumnos de ambos sexos mayores a 16 años para conocer los antecedentes patológicos y tóxicos, se cuestionó sobre la sintomatología y la frecuencia con la que realizaban actividad física, esto para conocer los problemas que presentan y los factores que la provocan.	Generar mayor conciencia entre los individuos para que acuda con regularidad al médico para estar en constante revisión y evitar o controlar futuras complicaciones en la salud que pongan en riesgo la calidad de vida.
de Paula Cerqueira, D., Tavares, J. R. & Machado, C. R., 2014. Factores de predicción de la insuficiencia renal y el algoritmo de control y tratamiento. <i>Revista Latino-Americana de Enfermagem</i> , abril, Volumen 2, pp. 211-217.	La insuficiencia renal es un padecimiento que se caracteriza porque la función renal disminuye y se acumulan metabolitos en el organismo, el desequilibrio lleva a acumular sustancias como urea y creatinina en la sangre. Los grupos con mayor probabilidad de padecer enfermedades renales son aquellos que padecen o tienen antecedentes familiares con diabetes, hipertensión, enfermedades cardiovasculares, los de raza negra y aquellos que padecen alguna otra enfermedad renal.	Se desarrolló un instrumento de recolección de datos, el cual estaba dividido en dos partes, la primera parte contiene antecedentes y hábitos de los pacientes, la segunda está formada por datos clínicos, tratamientos y pruebas de laboratorio, todos estos estudios se realizaron con un seguimiento riguroso para poder conocer el número de personas que desarrollaron insuficiencia renal, los factores que provocan dicha enfermedad y la evolución de ésta.	Realizar mayor número de estudios a todas las personas que así lo requieran, para que de esta forma se pueda disminuir el índice de padecimientos. Permitir el tratamiento a tiempo de las personas que la padezcan.
Morais Arruda Costa, G. y otros, 2016. Calidad de vida en pacientes con insuficiencia reanal crónica en hemodiálisis. <i>Enfermería Global</i> , julio, Volumen 15, pp. 59-72.	Aún con los avances tecnológicos que se han desarrollado a lo largo de los años, no ha habido algo que permita asegurar o mejorar la calidad de vida de las personas con insuficiencia renal, ya que un paciente con esta enfermedad presenta diversas restricciones o limitantes. El tratamiento recomendado para contrarrestar los efectos de esta enfermedad es la diálisis.	Se realizó un cuestionario a los pacientes para conocer las condiciones de vida que tenían, los cuales fueron digitalizados para realizar un análisis estadístico. Esta encuesta determinó que la mayoría de las personas que están en hemodiálisis son hombres, sin embargo, la mayoría de ellos tienen una relación sentimental estable, lo cual permite que la calidad de vida de los pacientes mejore.	Generar tecnología que permita mejorar la calidad de vida de las personas con enfermedades renales, las cuales les permitan realizar sus actividades diarias. Generar nuevas alternativas que permitan contrarrestar el avance y desgaste que esta enfermedad representa.
Velasco Rodríguez, G., 2018. Mercadotecnia social: las aplicaciones	Existen muchas aplicaciones orientadas al campo de la salud, sin embargo, no todas son confiables, por tal motivo las	Se realizó una clasificación de las aplicaciones de acuerdo con su funcionalidad, sistema operativo, categoría, a quien va dirigida y el	Cada desarrollador debe tener en cuenta estos parámetros de evaluación a la hora de crear una aplicación de salud, debido a que

<p>móviles en el mercado sanitario. <i>Horizonte sanitario</i>, enero, Volumen 17, pp. 9-20.</p>	<p>aplicaciones de salud son evaluadas tomando en cuenta diversos parámetros que definirán el grado de certeza con el que cuentan las aplicaciones.</p>	<p>tipo de pago. Esta clasificación permitió visualizar que la mayoría de las aplicaciones existentes actualmente están dirigidas a los profesionales del área médica.</p>	<p>deben mostrar un alto grado de confiabilidad para que los usuarios tengan la certeza de que la información obtenida es correcta.</p>
<p>Lugo Reyes, S. O., Maldonado Colín, G. & Murata, C., 2014. Inteligencia artificial para asistir el diagnóstico clínico en medicina. <i>Revista Alegria México</i>, marzo, Volumen 61, pp. 110-120.</p>	<p>Se han desarrollado softwares aplicados a diagnosticar ciertos padecimientos en específico, la mayoría hacen uso del aprendizaje automático, son los sistemas que pueden “aprender” a partir de datos iniciales como las redes neuronales, también están enfocados a la minería de datos, al mezclar las técnicas se puede obtener una mayor exactitud a la hora de predecir padecimientos.</p>	<p>Se han desarrollado diversos softwares que ayudan a los expertos a determinar un diagnóstico, tomar la decisión para prescribir un tratamiento y hacer un pronóstico de los resultados que se pueden obtener ante un tratamiento o terapia física.</p>	<p>Plantear las bases para el desarrollo de nuevas aplicaciones orientadas al campo de la salud, para hacer que éstas sean confiables para todos aquellos pacientes o personal médico que haga uso de ella. Desarrollo de nuevas tecnologías que puedan estar al alcance de la población que les permita llevar un manejo adecuado de sus enfermedades y tratamientos.</p>
<p>Bracamonte Alaniz, X., 2008. La Inteligencia Artificial en la Medicina. <i>Revista de Información, Tecnología y Sociedad</i>, Volumen 1, pp. 38-39.</p>	<p>En el campo de la salud han definido a la inteligencia como la facultad que se tiene para pensar, comprender y aprender de una situación en particular, es por ello por lo que se dice que pueden “pensar” debido a que se han desarrollado sistemas que ayudan a procesar gran cantidad de información.</p>	<p>Se determinó después de varios estudios que la prueba de Turing es capaz de determinar si está teniendo una conversación con un humano o una máquina, basándose en las respuestas recibidas. Todas las aplicaciones con inteligencia artificial deben ser analizadas por esta prueba para conocer la validez.</p>	<p>Desarrollo de aplicaciones para el campo de la salud, teniendo en cuentas las actuales y su funcionamiento, para generar mejoras o crear nuevas que estén enfocadas a otras aplicaciones dentro de la misma área de salud.</p>
<p>Alonso Arévalo, J. & Mirón Canelo, J. A., 2017. Aplicaciones móviles en salud: potencial, normativa de seguridad y regulación. <i>Revista Cubana de Información en Ciencias de la Salud</i>, 22 septiembre, Volumen 3, pp. 1-13.</p>	<p>La población que hoy en día hace uso de las aplicaciones de salud no siempre tiene total confianza en estas y por ello dejan de utilizarlas, por ello se deben realizar encuestas o algún otro método que permita conocer las opiniones de los usuarios y así mejorar la aplicación para que pueda ser utilizada con mayor provecho.</p>	<p>Se determinaron normativas para la evaluación de aplicaciones orientadas al campo de la salud, que permitirán hacer el análisis de la funcionalidad, manejabilidad y confiabilidad de todos los softwares que se quieran presentar tanto a los pacientes como a los expertos médicos.</p>	<p>Presentar los objetivos que se desean alcanzar en el desarrollo a futuro de aplicaciones para la salud, así como indicar las necesidades que se han ido generado a lo largo del tiempo para que futuros desarrolladores busquen la satisfacción de esta.</p>
<p>Beltrán Ramírez, R., Maciel Arellano, R. & Jiménez Arévalo, J., 2014. La tecnología y la inteligencia artificial como</p>	<p>Después de la creación de diversas herramientas de apoyo en el área de salud, se ha llegado a la conclusión de que la inteligencia artificial ha sido la pieza clave para el mejoramiento en la calidad de los</p>	<p>Se realizaron análisis de las técnicas de inteligencia artificial, las cuales debían ayudar a satisfacer las necesidades de los desarrolladores, por lo cual, se llegó a la conclusión de que los sistemas expertos,</p>	<p>Presentar los avances que ya se tienen para incentivar a los desarrolladores a crear más tecnología que permita satisfacer otro tipo de necesidades que aún no están cubiertas, tomando en cuenta los beneficios que se</p>

<p>futuro en el área médica. <i>Universitas Revista de ciencias sociales y humanas</i>, 15 diciembre, Volumen 21, pp. 185-190.</p>	<p>servicios que se prestan en hospitales y laboratorios. Los sistemas expertos han permitido el almacenamiento y procesamiento de información para la toma de decisiones.</p>	<p>minería de datos, redes neuronales y redes bayesianas eran la mejor opción para dicho trabajo.</p>	<p>tienen y los que se desean alcanzar con las nuevas aplicaciones que surjan con el paso del tiempo.</p>
<p>Instituto Mexicano del Seguro Social, s.f. Prevención, diagnóstico y tratamiento de la enfermedad renal crónica temprana. <i>Guía de referencia rápida</i>, Volumen 1, pp. 1-20.</p>	<p>Se determinan los factores de riesgo a considerar para diagnosticar insuficiencia renal, se mencionan los estudios clínicos que se deben realizar de rutina, así como los tratamientos que pueden suministrarse.</p>	<p>Se genera una lista de estudios médicos que se realizaran y generan los parámetros que darán como resultado los valores a analiza para determinar la enfermedad y el grado de avance que tienen.</p>	<p>Mejorar los estudios clínicos que se deben realizar a las personas para que puedan ser</p>
<p>Barrio, V., 2012. Necesidad y utilidad del empleo de criterios estandarizados para el diagnóstico de la disfunción renal aguda en pacientes críticos. <i>Medicina intensiva</i>, 36(10.1016), pp. 247-249.</p>	<p>La determinación de criterios para evaluar el grado de severidad que presenta una enfermedad renal es de vital importancia para los especialistas ya que mediante estos pueden definir el nivel de riesgo en el que se encuentra el paciente.</p>	<p>Múltiples organizaciones de salud han planteado diversos parámetros a considerar para las personas que padecen insuficiencia renal aguda, los cuales le permitirán conocer las posibles mejoras que tendrá un paciente a la hora de aplicar tratamiento.</p>	<p>Verificar si los elementos que se tienen actualmente son los adecuados y suficientes, de no ser así, mejorarlos para la generación del diagnóstico y ser más precisos, de esta forma permitir que se realice el tratamiento lo antes posible.</p>
<p>Bonilla, D., 2012. Algoritmos evolutivos en el diagnóstico de tumores de mama. <i>Seminario de ingeniería biomédica</i>, pp. 1-5.</p>	<p>Existen muchas aplicaciones actualmente que implementan técnicas de inteligencia artificial para determinar si un tumor es maligno o no, en este caso particular se hizo uso de algoritmos evolutivos que permitirán la clasificación de las muestras de tumores.</p>	<p>Se tomaron muestras a personas con indicios de padecer cáncer de mama, se analizaron y se evaluaron los parámetros en el algoritmo genético para determinar el estatus de los tumores.</p>	<p>Posibles mejoras en el algoritmo para que se presenten otras alternativas en la clasificación, que permita tener mayor precisión.</p>
<p>Delsol, K., Isaia, M. & Minetti, G., 2015. Aplicación de algoritmos aproximados al diagnóstico/clasificación de enfermedades. <i>Símpoio Argentino de Inteligencia</i></p>	<p>Se hizo uso de perceptrón multicapa algoritmos genéticos y algoritmos metaheurísticos híbridos para la clasificación de tumores cancerígenos y para determinar si tienen diabetes tipo II.</p>	<p>Se tomaron muestras a personas propensas, los valores fueron analizados y procesados en el algoritmo híbrido. Se desarrollaba el algoritmo genético y al momento de la mutación se realizaba mediante MLP.</p>	<p>Mayor profundidad en la investigación sobre las técnicas, para mejorar los resultados o perfeccionar las técnicas ya existentes.</p>

Artificial, 16(2451-7585), pp. 49-56.

Martínez Castelo, A. y otros, 2014. Documento de consenso para la detección y manejo de la enfermedad renal crónica. <i>Revista nefrología. Órgano oficial de la sociedad española de nefrología</i> , 34(0212-6567), pp. 243-62.	Se hace uso de los criterios KDIGO para la determinación de los parámetros en la detección de insuficiencia renal, hace comparaciones con otros algoritmos para determinar cuál de ellos es la mejor opción a utilizar en el diagnóstico de esta enfermedad.	Determinación de algunos factores de riesgo en las enfermedades renales, cada uno de ellos tienen elementos individuales que se deben considerar para determinar los límites en el que se encuentran los pacientes y definir la situación médica.	Tener mayor definición de los niveles a considerar de cada uno de los factores de riesgo para poder facilitar el análisis de las pruebas realizadas a los pacientes y mejorar el diagnóstico proporcionado por el médico especialista.
Villamizar Suaza, K., Soto Durán, D. E., Giraldo Mejía, J. C. & Jiménez Builes, J. A., 2016. Modelo de pruebas en proyectos BI. <i>Engineering Innovations for Global Sustainability</i> , 14(2414-6390), pp. 1-9.	Definición del modelo en V, adaptación de este al proyecto en cuestión y descripción de cada una de las fases que se deben llevar a cabo con el desarrollo de esta metodología.	Define cada una de las fases que tiene el método para poder llevar a cabo las tareas que comprende cada fase y así asegurar el correcto funcionamiento del software.	Utilización del modelo en V de acuerdo con las necesidades del usuario, si este método las cumple, se pueden adaptar a cada uno de los proyectos que se deseen realizar para que se logren con éxito.

5 MÉTODO

En esta sección se presenta el desarrollo procedimental del diseño, experimentación y validación de la aplicación desarrollada, así como del sistema de inferencia. Se destaca el uso de una metodología de desarrollo software en V, ya que permite una retroalimentación de cada una de las etapas de diseño a las fases de implementación.

5.1 Procedimiento de diseño e implementación de la aplicación

En la Figura 6 se presentan los pasos que se siguieron durante el desarrollo de la aplicación móvil para apoyo al seguimiento y detección de riesgos en pacientes con insuficiencia renal crónica, consistió en identificar la mejor técnica de softcomputing, esta fue codificada en la plataforma Android, la cual permitió el desarrollo del software, se realizó un único algoritmo mediante lógica difusa, este es capaz de proporcionar un porcentaje estimado de los riesgos que tiene una persona de padecer alguna enfermedad renal, esto se logró mediante la determinación de los parámetros a considerar para la generación del diagnóstico, como son hábitos alimenticios, rutina diaria, antecedentes familiares, esto haciendo referencia a enfermedades que se han desarrollado como diabetes, hipertensión arterial, obesidad, entre otras.

Se llevaron a cabo las pruebas de validación correspondientes a cada una de las etapas de la metodología en V que se siguió durante el proceso de desarrollo de la aplicación, esto mediante la comparación de los datos proporcionados por la aplicación y el diagnóstico que facilitó el médico experto, esto se realizó con la finalidad de asegurar que la información emitida por el sistema fuera el correcto para que todos los usuarios tengan la certeza de que los datos son los más acertados de acuerdo con la información que ellos generen y que será procesada por el algoritmo, cuando se presentaron algunos problemas con la asertividad del algoritmo se rediseño, una vez que se tuvo un mejor desempeño del algoritmo, se procedió a realizar la documentación de resultados en el documento de tesis.

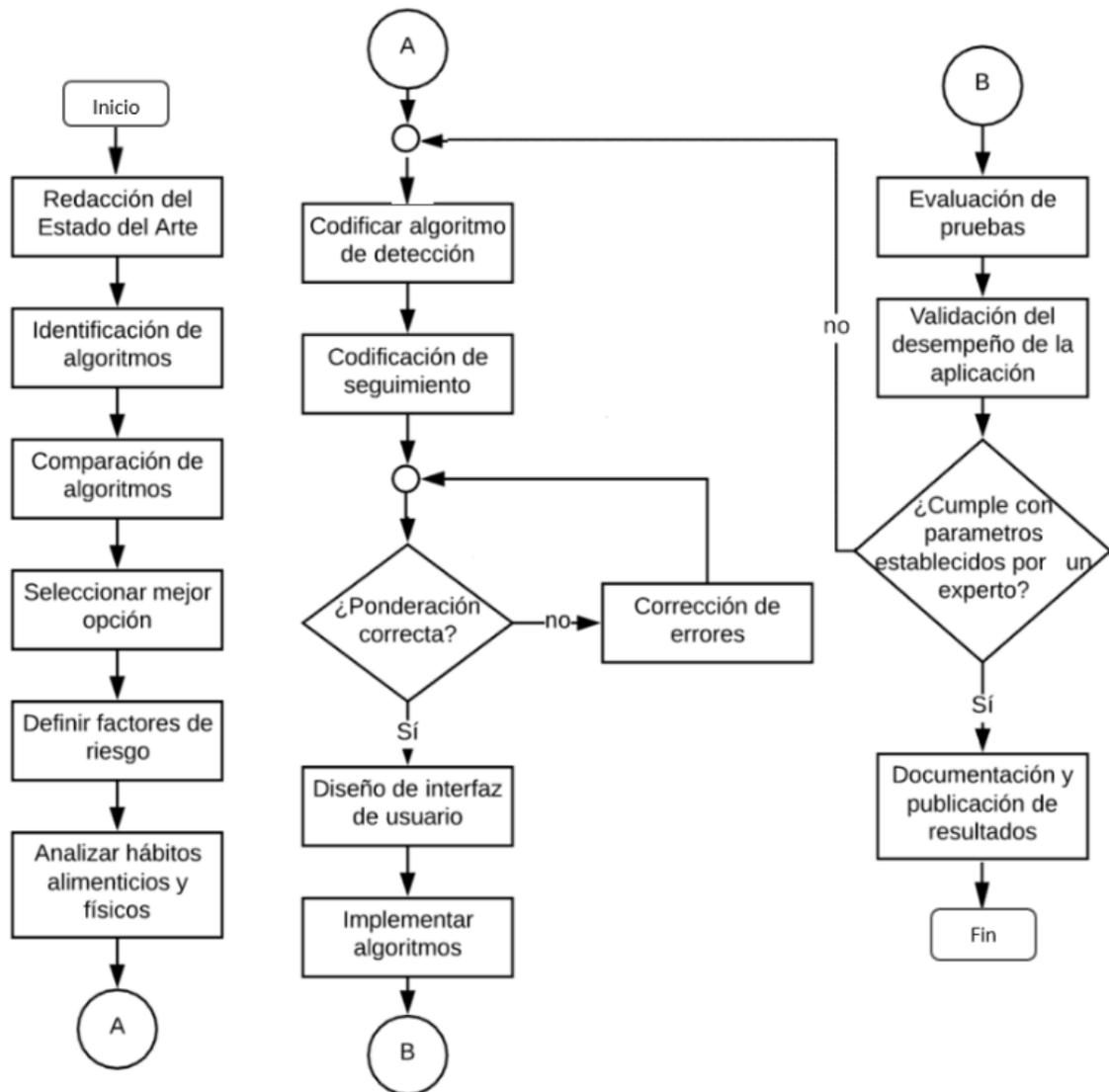


Figura 6 Diagrama de flujo del proyecto.

5.1.1 Modelo de desarrollo de software

Para poder desarrollar un software existen diversas metodologías que son utilizadas para tener una estructura y controlar el proceso de crear un sistema. Seguir una metodología otorga mayores posibilidades de tener éxito durante la creación del software, gracias a que se establece una serie de tareas que se van completando desde que se plantea la idea de un software, hasta el término de este. Las diferentes metodologías tienen como objetivos en común, los siguientes (Cuma, 2017):

1. Uniformidad y calidad en el desarrollo del sistema.

2. Cumplir con las necesidades del usuario al que será dirigido el sistema.
3. Tener mejor eficiencia y rendimiento en cuanto a los desarrolladores.
4. Cumplir con los plazos y costes.
5. Facilitar el mantenimiento del sistema.
6. Definir los puntos de control y revisión.
7. Facilidad para documentar el sistema.
8. Identificar con mayor facilidad algún cambio que se requiera durante el desarrollo para el mejoramiento del software.

El desarrollo de esta aplicación móvil se ha desarrollado siguiendo el modelo en V, que es una metodología para el desarrollo de software que representa el flujo de actividades que se deben llevar a cabo para la creación de programas de cómputo (Perez, et al., 2006). Tiene la representación de una “V” debido a que se realizan en paralelo las fases de construcción y la fase de pruebas, así como se muestra en la Figura 7.

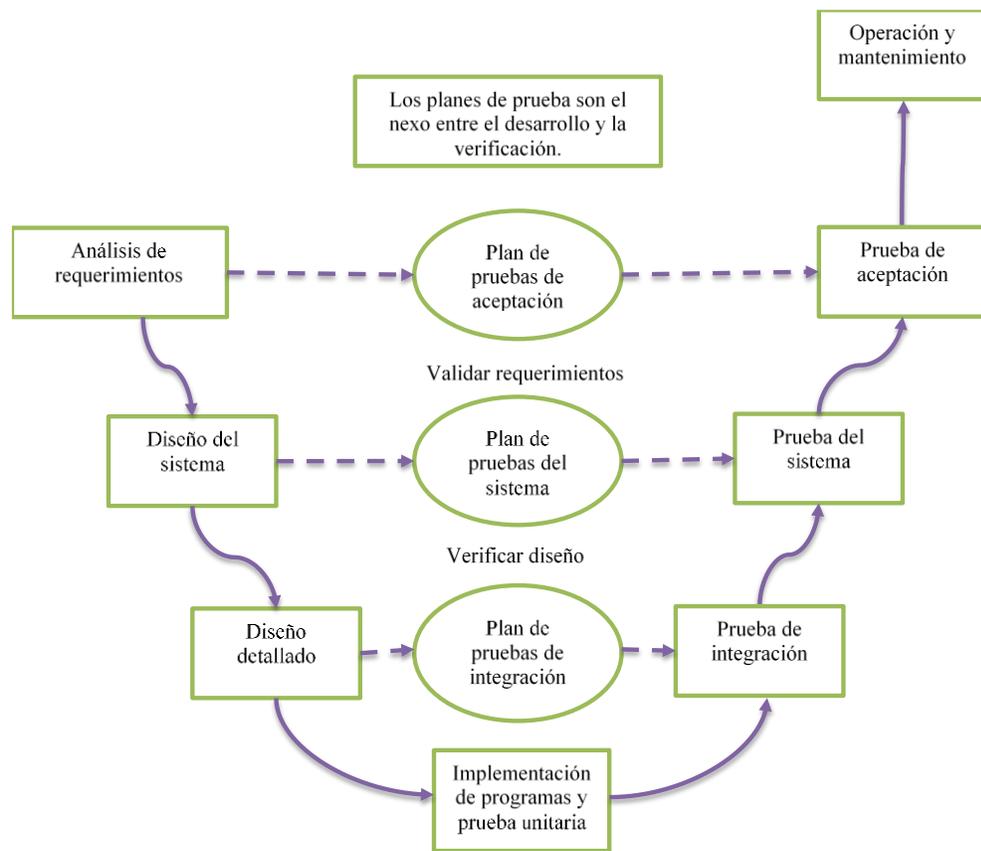


Figura 7 Modelo en V para el desarrollo de software.

El modelo seleccionado cuenta con 4 niveles lógicos, cada uno de ellos tiene una fase paralela de pruebas, los niveles consisten en:

Nivel 1: Tiene inclinación a los usuarios del sistema, ya que se determina que se requiere solucionar y de qué manera y la validación del producto final, lo cual realizarán las personas a las que va dirigido el software.

Nivel 2: Hace referencia a las características funcionales que debe cumplir el sistema.

Nivel 3: Se trata de la definición del software y hardware que se requerirán para el producto final, a esto se le llama arquitectura del sistema.

Nivel 4: Es la etapa de codificación, se desarrollan los módulos que harán funcional el sistema.

Algunas ventajas que ofrece este modelo es que la relación entre las etapas de desarrollo y las pruebas facilitan la identificación de fallas en el funcionamiento, es un modelo sencillo de implementar, hace más ágil el proceso de revisión del sistema, especifica bien el objetivo de cada una de las pruebas que se realizan e involucra siempre a los usuarios en cada una de las pruebas para determinar si es funcional para ellos y se hacen modificaciones a la estructura de no ser así (Villamizar Suaza, et al., 2016).

5.1.2 Sistemas gestores de bases de datos para aplicaciones móviles

Actualmente existen diversos tipos de herramientas que permiten el manejo de una gran cantidad de información; cada una de ellas cuenta con características específicas que se adecúan a las necesidades de los usuarios. Estas herramientas son los Sistemas Gestores de Bases de Datos (*SGBD*), donde un gestor es aquel que se encarga de la interfaz existente entre el usuario y los datos.

Para desarrollar una base de datos en Android es necesario tener en cuenta que deben ser ligeras, debido al limitado almacenamiento con el que cuentan los dispositivos móviles, sin necesidad de un servidor, debe estar disponible en el momento que se requiera sin tener una dependencia, deben ser rápidas y seguras con la información que se está

almacenando, tienen que ser de fácil control y manejo, debe tener un consumo bajo de energía y de espacio de memoria.

Existen diversos SGBD para el sistema operativo Android, el más utilizado según (M., 2018) es SQLite, una de las características de este motor es que puede almacenar información y recuperarla de manera sencilla, cuenta con diversos enlaces a distintos lenguajes de programación como Java, C, C++, JavaScript, C#, Python, VB Script, etc. Los tipos de datos que maneja son int, varchar, blob, real, double, float, text, boolean, date, datetime, entre otros.

Algunas ventajas que tiene la utilización de este SGBD son las siguientes:

1. No requiere de configurar rutas, tamaños, puertos, etc.
2. Cuenta con librerías que se encargan de gestionar, lo que significa que no se ejecutan procesos para administrar información.
3. Almacena toda la información en un archivo multiplataforma, por lo que simplifica las copias de seguridad y la migración de datos.
4. Permite que los datos queden almacenados aun cuando el dispositivo este apagado.
5. Reduce el tiempo de inicio y el consumo de memoria ya que solo carga los datos que se necesiten, en lugar de leer el archivo completo.
6. El contenido se actualiza constantemente para evitar la pérdida de información.

Las bases de datos creadas en Android únicamente son visibles para la aplicación para la que fueron creadas.

Para este proyecto particular, se hizo uso del SGBD llamado SQLite, debido a que es un motor de base de datos SQL autónomo y que no requiere de la conexión a un servidor. Es el más implementado a nivel mundial (SQLite, 2019). Una de las principales características por las cuales se utilizó en el proyecto es que no requiere de un servidor separado ya que lee y escribe directamente en la memoria del móvil, lo que lo hace de fácil y rápido acceso, sin la necesidad de tener una conexión a internet, de igual forma por la accesibilidad que se tiene de los datos almacenados.

EL tamaño de la biblioteca SQLite puede llegar a ser inferior a 600KB, por lo que no representa un problema para ser almacenada en un teléfono móvil, el rendimiento es bueno

aun cuando la memoria que se le dé sea poca. Es uno de los SGBD más utilizados para el desarrollo de aplicaciones.

5.1.3 Diseño de la interfaz gráfica

Para la determinación de las funciones que tendrá la aplicación móvil, se hizo uso de un mapa de navegación, estos mapas permiten analizar que secciones del producto faltan y cuales sobran, establecen la relación entre los contenidos de acuerdo con su afinidad y jerarquía, así como también muestran que secciones podrían ampliarse en próximos trabajos. En la Figura 8 se presenta el mapa de navegación que se desarrolló para la aplicación móvil.

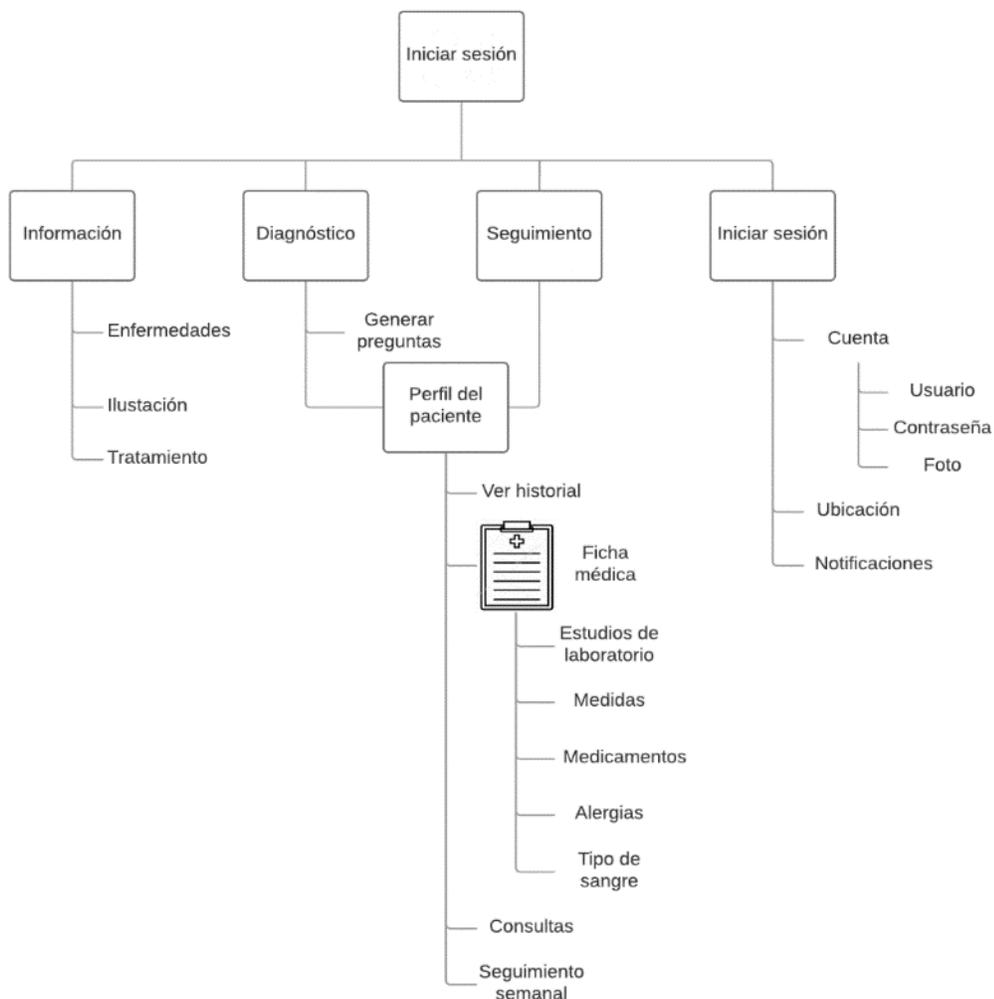


Figura 8 Mapa de navegación para la aplicación móvil.

Para tener un panorama más amplio de las vistas que tendrá la aplicación móvil, se hizo uso de los *wireframes*, que son aquellas representaciones del diseño de cada una de las pantallas que van a conformar la aplicación. A continuación, se muestra el *wireframes* empleado para este trabajo de investigación.

Al descargar la aplicación móvil se inicia una guía, la cual permite dar al usuario toda la información necesaria para que conozca el funcionamiento de esta, en la Figura 9 se da una explicación general del objetivo principal que tiene la aplicación, en la Figura 10 se explica de qué manera funciona el apartado de seguimiento, en la Figura 11 se muestra el manejo de las preguntas que permite hacer la evaluación para determinar el porcentaje de riesgo de padecer una enfermedad renal, finalmente, en la Figura 12 se describen los términos y condiciones que tiene el uso de la aplicación.

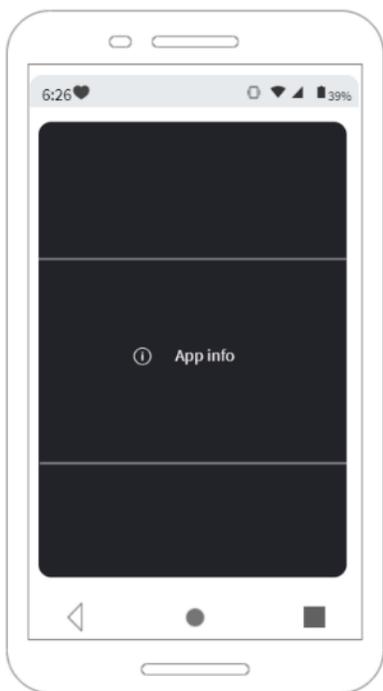


Figura 9 Información general sobre el funcionamiento de la aplicación.

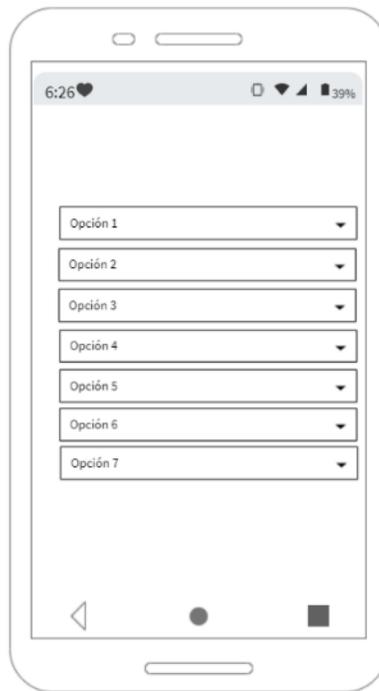


Figura 11 Funcionamiento del apartado de seguimiento.

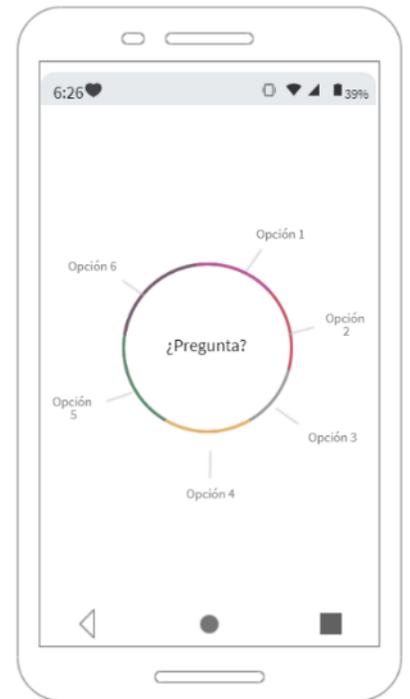


Figura 10 Funcionamiento del apartado de diagnóstico.

Una vez culminada la guía de funcionamiento, se procederá a iniciar sesión (Figura 13) o al registro del usuario (Figura 14), esto permitirá que los datos ingresados en cada uno de los apartados de la aplicación estén debidamente resguardados mediante la contraseña que autenticará al usuario, que será el único con acceso a visualizar, modificar o eliminar los datos de su cuenta.



Figura 13 Términos y condiciones que tendrá el uso de la aplicación.

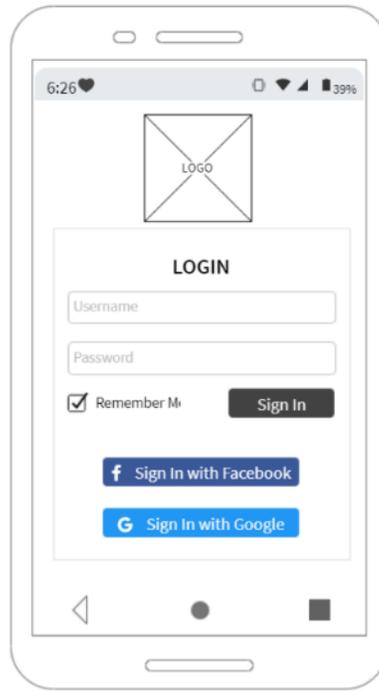


Figura 14 Inicio de sesión.

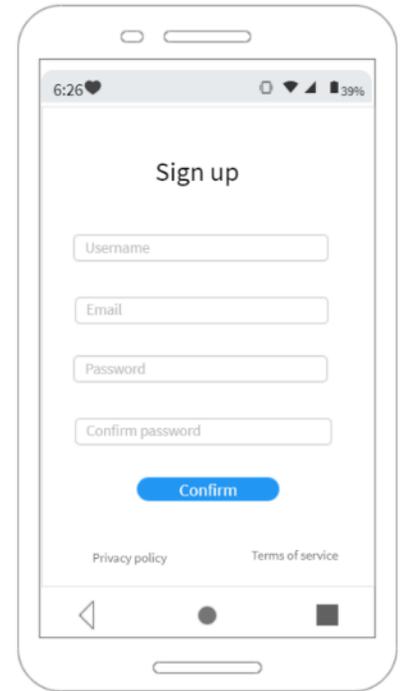


Figura 12 Registro del usuario.



Figura 15 Menú principal.

Al ser autenticado el usuario, se tendrá acceso a un menú principal (Figura 15), en el cual, el usuario seleccionará mediante botones el apartado que desea visitar.

En el apartado de información, se mostrará una pantalla en la que se enliste la clasificación de las enfermedades renales, tal como se muestra en la Figura 16, una vez que el usuario seleccione una opción, se mostrará toda la información relacionada con dicha enfermedad (Figura 17), posteriormente, podrá elegir entre dos opciones, la primera (Figura 18), mostrará una comparación del órgano saludable y uno que padece la enfermedad, y la segunda (Figura 19) proporcionará información sobre el tratamiento a seguir para tratar con la enfermedad.

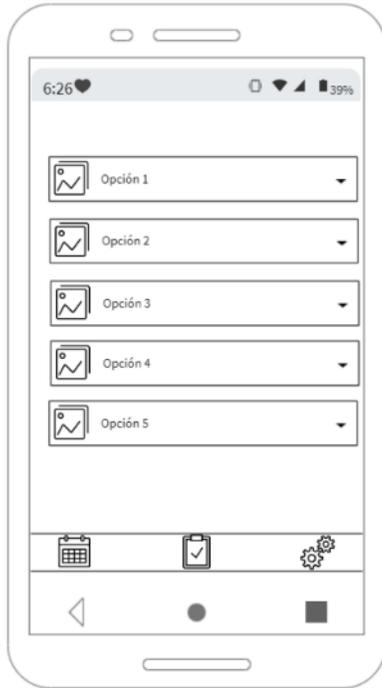


Figura 18 Clasificación de enfermedades renales.

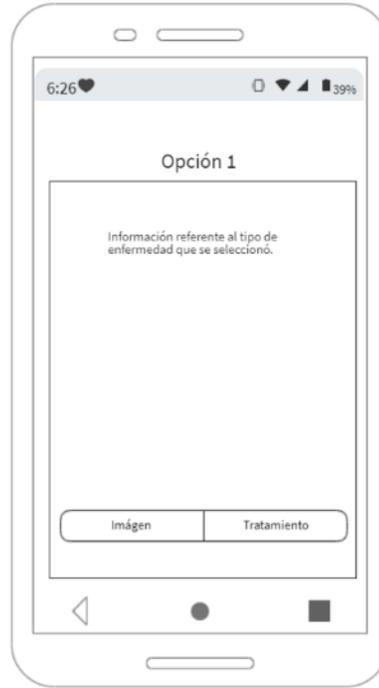


Figura 17 Información de la enfermedad.

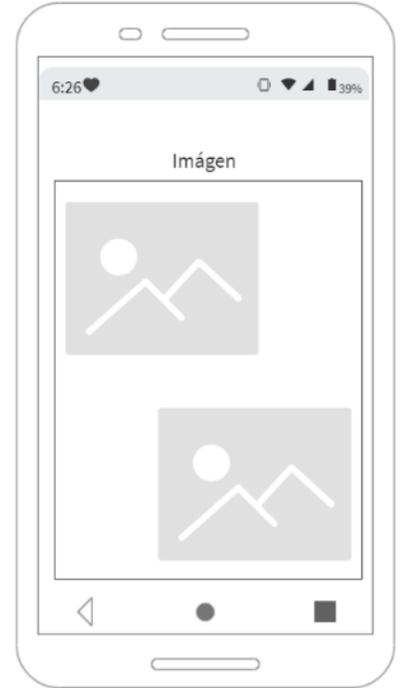


Figura 16 Comparación del órgano.

En el diagnóstico, se generarán una serie de preguntas como se muestra en la Figura 20, las cuales deberán ser respondidas por el usuario, estas respuestas se evaluarán y permitirán determinar el porcentaje de riesgo que tiene dicho usuario (Figura 21).



Figura 21 Información sobre el tratamiento a seguir.

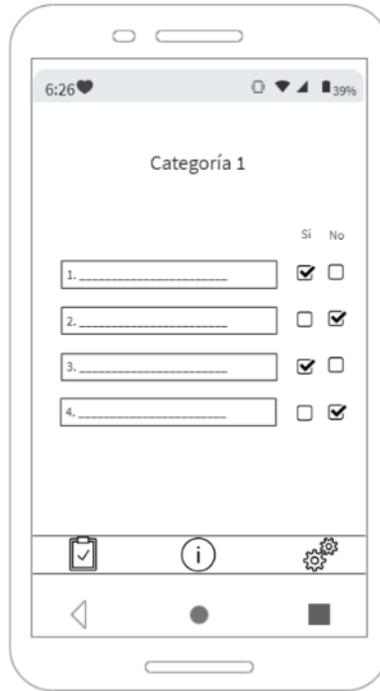


Figura 19 Preguntas generadas para el diagnóstico.



Figura 20 Determinación del porcentaje de riesgo.

Se tendrá un apartado de expediente como se observa en la Figura 22, en el cual, el usuario colocará su información en todos los apartados que así lo requieran, esto para poder generar el avance que se tiene semanalmente y pueda ser mostrado al usuario mediante una gráfica.

Finalmente se tendrá una ventana en la que se muestren los ajustes (Figura 23), aquí el usuario podrá modificar el nombre de usuario, la contraseña y la foto de perfil, de igual manera se podrá activar o desactivar la opción de recibir notificaciones y permitir el acceso a la ubicación.

En los diagramas de caso de uso, un requerimiento que ha sido solucionado por el sistema es representado por una elipse, el conjunto de todas ellas representa la totalidad de operaciones que se llevaran a cabo dentro de la aplicación móvil.



Figura 23 Perfil del usuario.



Figura 22 Ajustes de la aplicación.

Para desarrollar la fase de análisis del sistema, se diseñó el diagrama de caso de uso que se muestra en la Figura 24, en el cual se muestran las operaciones de la aplicación móvil y la forma en que interactúa con el usuario.

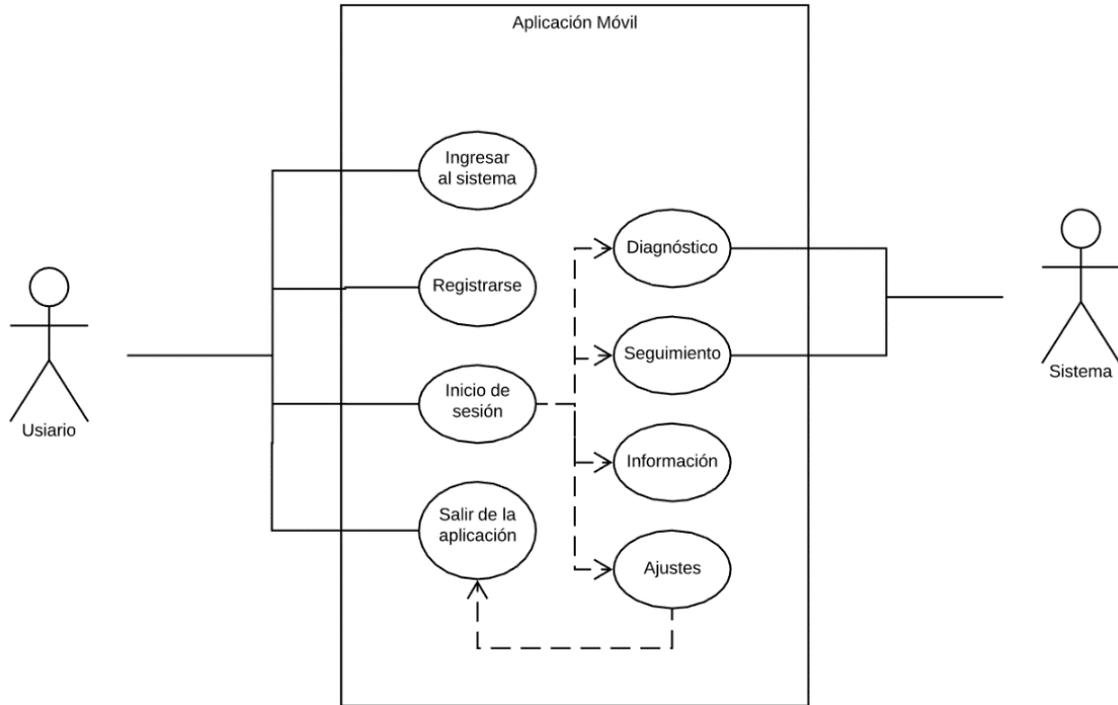


Figura 24 Diagrama de casos de uso.

Para describir la estructura del sistema, se hizo uso de los diagramas de clases o UML, esto permitirá mostrar cada una de las clases que comprenden la aplicación, los atributos con los que cuenta cada una de ellas y la relación que existe. Estos diagramas son de gran utilidad en el proceso de análisis y diseño de un sistema ya que se conceptualiza la información que se maneja dentro del sistema.

Al utilizar el paradigma de programación orientada a objetos, los conceptos se incorporan en el diagrama de clases de la Figura 25.

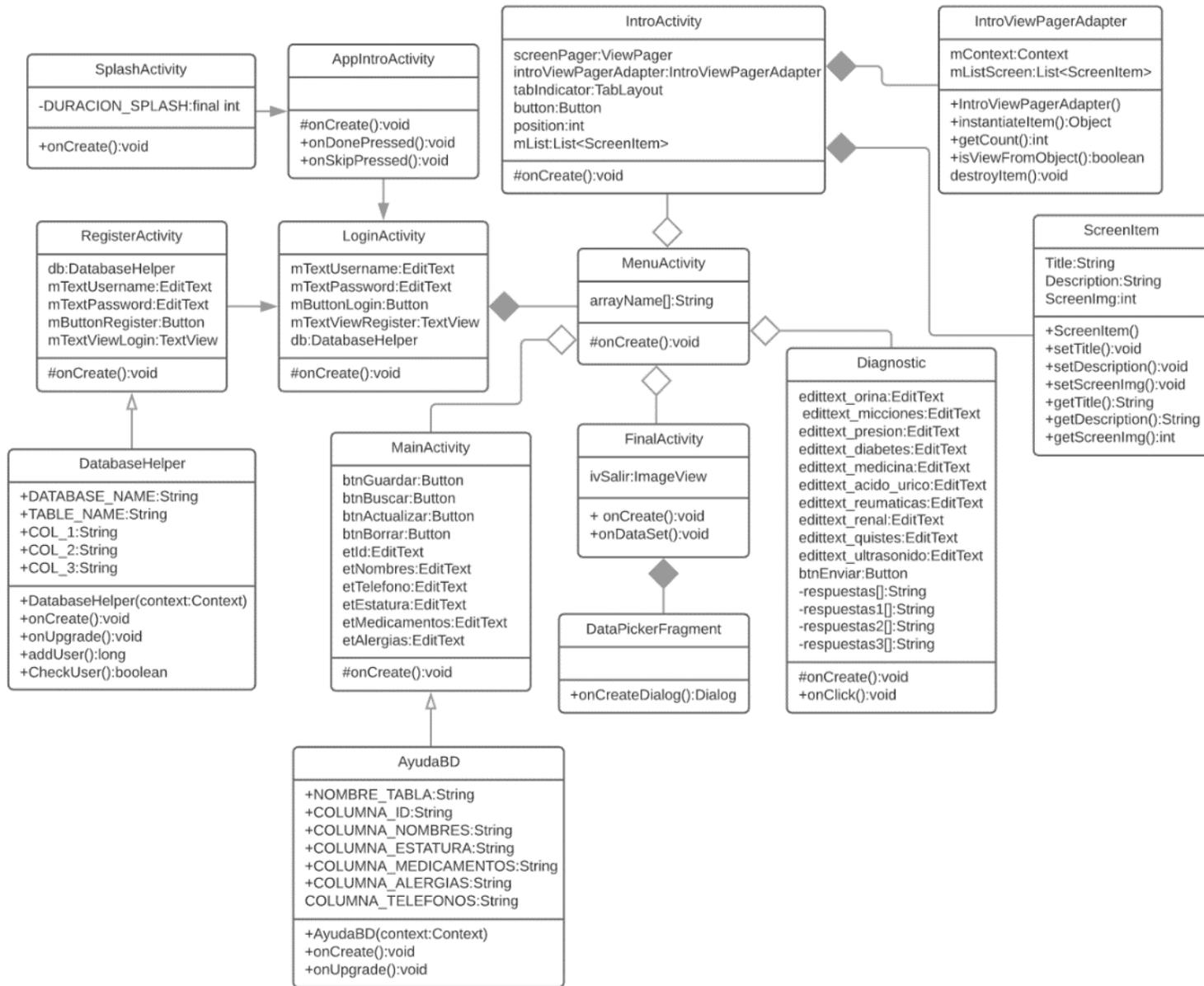


Figura 25 Diagrama de clases de la aplicación móvil.

5.1.4 Programación de la aplicación

El paradigma de programación que se utilizó para el desarrollo de la aplicación es orientado a objetos ya que permite descomponer el problema en pequeños subgrupos y de esta forma facilitan la construcción de sistemas complejos.

Android Studio es una plataforma que permite desarrollar este paradigma, como todo proyecto desarrollado en Android, el proyecto está dividido en dos directorios principales: **java**, que como su nombre lo indica, es el código fuente en java de la aplicación y **res**, que son los ficheros XML que definen la interfaz gráfica de la aplicación.

Para esta aplicación, se desarrollaron nueve clases, las cuales se describen en la Tabla 4, de acuerdo con el diseño que se ha descrito en la sección anterior.

Tabla 4. Clases del paquete app.

Clase	Descripción
RegisterActivity	Esta clase toma la información ingresada por el usuario en las cajas de texto y las hereda a la clase DatabaseHelper para crear la cuenta de usuario, también se define la acción que realizará el botón de la interfaz.
DatabaseHelper	En esta clase se crea la base de datos en SQLite, en donde se almacenan el nombre de usuario, la contraseña y la confirmación de esta para que el usuario tenga acceso a todas las funciones del sistema.
LoginActivity	Se determina si el usuario y la contraseña han sido previamente almacenados en la base de datos, de ser así permite el acceso a las demás funciones del sistema, de lo contrario mostrará un mensaje de error.
MenuActivity	Puede observarse un menú que se despliega después de presionar el círculo central, en él se muestran las opciones de diagnóstico, seguimiento, información y ajustes.
MainActivity	Se determinan las acciones que realiza cada uno de los elementos colocados en la interfaz gráfica, estos datos son heredados posteriormente a la clase AyudaBD.
AyudaBD	Se crea una tabla en la base de datos SQLite, que permite almacenar la información recopilada por la clase MainActivity, los cuales son: el id, peso, estatura, medicamentos, alergias y tipo de sangre del usuario.
FinalActivity	Permite que el usuario pueda salir de manera exitosa de la aplicación, garantizando que los datos que ingresó durante la sesión han sido debidamente almacenados y estarán disponibles en el siguiente inicio de sesión, así como seleccionar la fecha de la próxima cita.

IntroActivity	Despliega la información acerca de las enfermedades renales que existen, así como una imagen representativa del padecimiento.
IntroViewPagerAdapter	Hereda atributos de la clase IntroActivity, se define la posición en donde se coloca el título y la descripción de la enfermedad.
ScreenItem	Se determina la imagen representativa, así como el diseño de la vista que interactúa directamente con el usuario.
SplashActivity	Permite visualizar el logo de la aplicación como vista principal.
Diagnostic	Contiene una serie de preguntas que serán contestadas por el usuario, posteriormente al ser presionado el botón de Enviar se evaluarán las respuestas mediante lógica difusa, asignando de esta manera un porcentaje de riesgo de padecer una enfermedad renal.
AppIntroActivity	Presenta tres vistas de inicio que dará información al usuario acerca del objetivo de la creación de la aplicación, así como informar que no se pretende reemplazar la valoración de un médico especialista.
DatePickerFragment	Permite desplegar el calendario para que el usuario seleccione la fecha de su próxima cita.

5.2 Sistema de inferencia

5.2.1 Definición de conjuntos difusos

Los conjuntos difusos fueron introducidos por Lotfi A. Zadeh en 1965, fueron una extensión de los conjuntos clásicos, estos permitían un grado de pertenencia gradual de cada uno de los elementos al conjunto.

Una variable lingüística es aquella cuyos valores son sentencias del lenguaje natural en vez de valores numéricos. Cada uno de los valores se denominan términos o etiquetas lingüísticas. Cada una de las variables lingüísticas tiene una regla semántica que asocia a cada etiqueta un número difuso, el cual determina el grado de compatibilidad de x con el conjunto.

Por ejemplo, para una variable lingüística que puede ser rendimiento, las etiquetas lingüísticas serían “muy bajo”, “bajo”, “medio”, “alto” y “muy alto”, cada una de estas etiquetas se asocian con un número o restricción difusos, lo que hace diferente a los conjuntos difusos de los clásicos, es que en el primero un mismo valor puede pertenecer

a diferentes etiquetas lingüísticas al mismo tiempo. En la Figura 26 se muestra la representación del conjunto difuso.

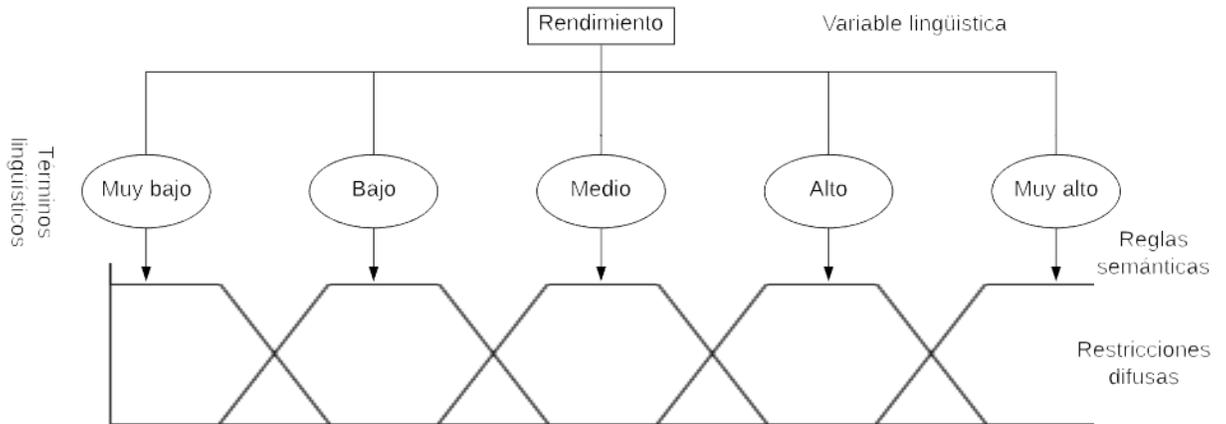


Figura 26 Variable lingüística y restricciones difusas asociadas (Sáa, 2012).

5.2.2 Función de inclusión

Las funciones de inclusión son las expresiones matemáticas que permiten determinar el grado de pertenencia de cada uno de los elementos que conforman un conjunto y pueden tener valores entre 0 y 1.

Una función de inclusión o pertenencia (*membership function*) de un conjunto difuso es un grupo de funciones definidas que, en la toma de decisiones mediante reglas expertas, suministra al motor de lógica difusa de entradas difusas. Las funciones más frecuentes son de tipo trapezoidal y triangular.

- Función trapezoidal: Definida por la ecuación 1.

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a < x \leq b \\ 1 & b < x \leq c \\ \frac{d-x}{d-c} & c < x \leq d \\ 0 & x > d \end{cases} \quad (1)$$

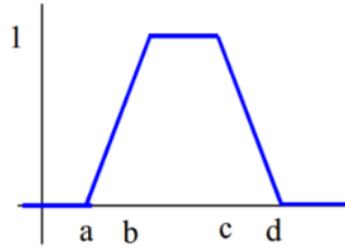


Figura 27 Función de pertenencia de tipo trapezoidal.

- Función triangular: Definida por la ecuación 2.

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{m-a} & a < x \leq m \\ \frac{b-x}{b-m} & m < x \leq b \\ 0 & x > b \end{cases} \quad (2)$$

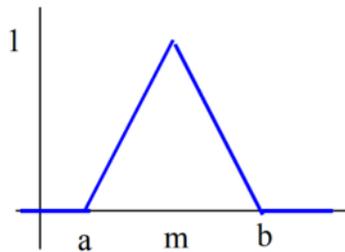


Figura 28 Función de pertenencia de tipo triangular.

- Función GAMMA (Γ): Definida por la ecuación 3.

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{m-a} & a < x < m \\ 1 & x \geq m \end{cases} \quad (3)$$

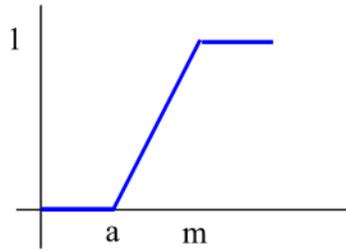


Figura 29 Función de pertenencia de tipo GAMMA.

- Función L: Esta definida como 1 menos la función GAMMA.

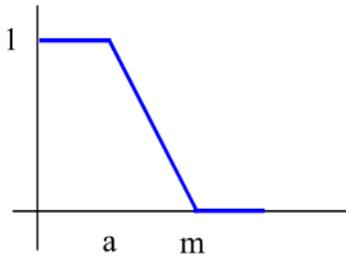


Figura 30 Función de pertenencia de tipo L.

- Función S: Definida por la ecuación 4.

$$\mu_S(x) = \begin{cases} 0 & x \leq a \\ 2\left(\frac{x-a}{c-a}\right)^2 & a \leq x \leq \frac{a+c}{2} \\ 1 - 2\left(\frac{x-a}{c-a}\right)^2 & \frac{a+c}{2} \leq x \leq c \\ 1 & x \geq c \end{cases} \quad (4)$$

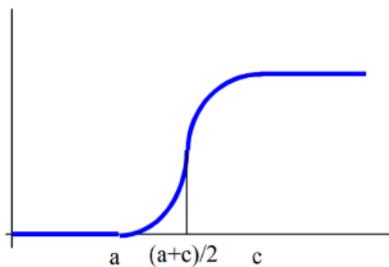


Figura 31 Función de pertenencia de tipo S.

- Función Z: Definida por la ecuación opuesta.

$$\mu_z(x) = 1 - \mu_s(x)$$

- Función Π : Definida por la ecuación 5.

$$\mu_{\Pi}(x) = \begin{cases} \mu_s(x) & x \leq b \\ \mu_z(x) & x > b \end{cases} \quad (5)$$

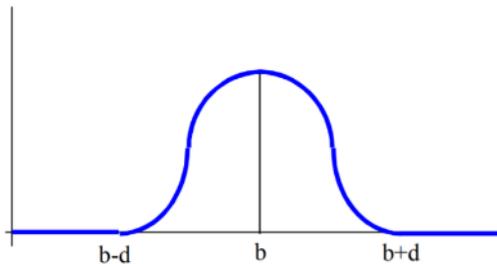


Figura 32 Función de pertenencia de tipo Π .

- Función Singleton: Definida por la ecuación 6.

$$\mu(x) = \begin{cases} 1 & x = a \\ 0 & x \neq a \end{cases} \quad (6)$$

5.2.3 Inferencia difusa

Es el proceso que permite obtener un único valor de salida para cada valor de entrada mediante la teoría de conjuntos difusos. Existen dos tipos de inferencia, el modelo de Mamdani y el de Takagi, Sugeno y Kang (TSK).

Inferencia de Mamdani

Es un método propuesto por Ebrahim Mamdani en 1975, se lleva a cabo con los siguientes pasos:

1. Fuzificación de variables de entrada: Consiste en tomar valores *crisp* (conjuntos clásicos) de las entradas y determinar el grado de pertenencia de las entradas que se tienen a los conjuntos difusos asociados.

2. Evaluación de reglas: Se aplican a los antecedentes de las reglas difusas las entradas, en caso de tener múltiples antecedentes, se realiza la operación AND u OR para obtener un único valor que lo representará. Para la evaluación se emplea la búsqueda del máximo en caso de realizar una operación OR y mínimo para la operación AND.
3. Integración de salidas de las reglas: Se unifican las salidas de cada una de las reglas, esto permitirá que se genere un único conjunto difuso para cada variable de salida.
4. Defuzificación: Se tiene como entrada el conjunto difuso obtenido en el paso anterior para generar un valor de salida.

Las reglas difusas son todas aquellas proposiciones *if-then* que permiten representar el problema a resolver, con ellas se formulan las condicionantes que se utilizan en la lógica difusa.

Existen sistemas difusos de tipo MISO (*Multiple Input Single Output*) el cual tiene n número de entradas y una única salida, los de tipo MIMO (*Multiple Input, Multiple Output*) y tienen múltiples entradas y múltiples salidas, el tipo SIMO (*Single Input Multiple Output*) tienen una única entrada y múltiples salidas, y finalmente el tipo SISO (*Single Input Single Output*) que tienen una entrada y una salida.

A continuación, se muestran algunos métodos de implicación para la interpretación de las reglas:

- Implicación difusa por regla de mínimos:

$$\mu_{A \rightarrow B}(x, y) = \min[\mu_A(x), \mu_B(y)] \quad (7)$$

- Implicación difusa para la regla del producto:

$$\mu_{A \rightarrow B}(x, y) = \mu_A(x)\mu_B(y) \quad (8)$$

- Implicación difusa para la regla aritmética:

$$\mu_{A \rightarrow B}(x, y) = \min[1, 1 - \mu_A(x) + \mu_B(y)] \quad (9)$$

- Implicación difusa para la regla Max-min:

$$\mu_{A \rightarrow B}(x, y) = \max\{\min[\mu_A(x), \mu_B(y)], 1 - \mu_A(x)\} \quad (10)$$

- Implicación difusa para la regla booleana:

$$\mu_{A \rightarrow B}(x, y) = \max\{1 - \mu_A(x), \mu_B(y)\} \quad (11)$$

Inferencia TSK

Este método, desde el punto de vista computacional no ha sido considerado como eficiente; es posible disminuir el tiempo de inferencia mediante una función matemática, la regla en inferencia TSK es:

$$p: \text{IF } x \text{ es } A \text{ AND } y \text{ es } B \text{ THEN } z \text{ es } f(x, y)$$

Una vez aplicada esta regla, el método se vuelve más eficiente, y la forma más común de representarla es mediante la función Singleton, la cual toma el valor uno en un valor específico dentro del universo de discusión y cero en cualquier otro punto (Morcillo, 2019).

En conclusión, la lógica difusa es un mecanismo de inferencia que permite simular el razonamiento humano, proporciona un contexto matemático que permite que la incertidumbre pueda ser formulada, de tal manera que permitan a una computadora tratar la información proporcionada.

5.3 Escala de Likert

Este tipo de escalas es ampliamente utilizado cuando se requiere la medición de respuestas, facilita la construcción y administración del cuestionario, de igual forma la codificación y análisis de la información que ha sido recabada (Maese Núñez, et al., 2016).

Se elabora una serie de ítems sobre el tema del que se realiza la investigación, los individuos deben elegir una de las respuestas posibles acorde a su grado de acuerdo o desacuerdo. Algunas de las ventajas del uso de esta escala son:

- Es fácil de construir, de comprender y contestar.

- Menor ambigüedad en las respuestas.

Algunas desventajas son:

- Reduce la descripción y diversidad de las respuestas.
- El análisis estadístico es reducido.
- La codificación de las respuestas no refleja las diferencias entre las respuestas.

Un ejemplo de conversión difusa de las respuestas obtenidas por la escala de Likert se muestra en la Figura 33. El empleo de números difusos permite representar la imprecisión de la escala Likert.

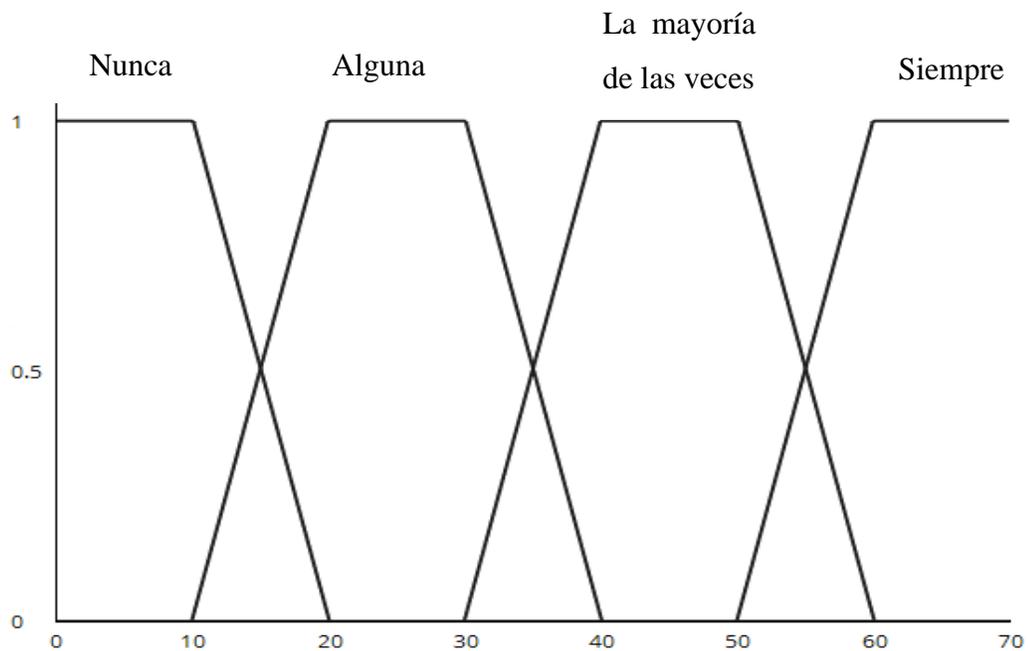


Figura 33 Conversión de la escala de Likert a Fuzzy (Sáa, 2012).

La escala de números Fuzzy es frecuentemente utilizado desde la ingeniería de control hasta el diagnóstico médico.

Para el desarrollo del sistema experto, se hará uso de Lógica Difusa, debido a que esta disciplina matemática es aplicada a conceptos que pueden tomar un valor cualquiera de veracidad dentro de un conjunto de valores que están entre dos extremos. Los sistemas

basados en lógica difusa combinan variables de entrada definidas en conjuntos difusos por medio de reglas que pueden producir uno o varios valores de salida.

Los conjuntos de entrada que se utilizarán en el desarrollo del proyecto son:

- Número de veces que orina en 24 horas.
- Número de micciones por la noche.
- Problemas de presión arterial.
- Diabetes.
- Exámenes prostáticos en caso de ser hombre mayor a 40 años.
- Ingerir medicamentos con frecuencia.
- Padecer ácido úrico.
- Enfermedades reumáticas.
- Familiares con enfermedades renales.
- Familiares con quistes renales.
- Realización de ultrasonidos.

Los conjuntos de entrada que se definieron para este proyecto son: Número de veces que orina en 24 horas (OD), micciones por la noche (MN) y presión arterial (PA).

El conjunto de entrada para el número de veces que orina en 24 horas $f(OD)$ se muestra en la Figura 34.

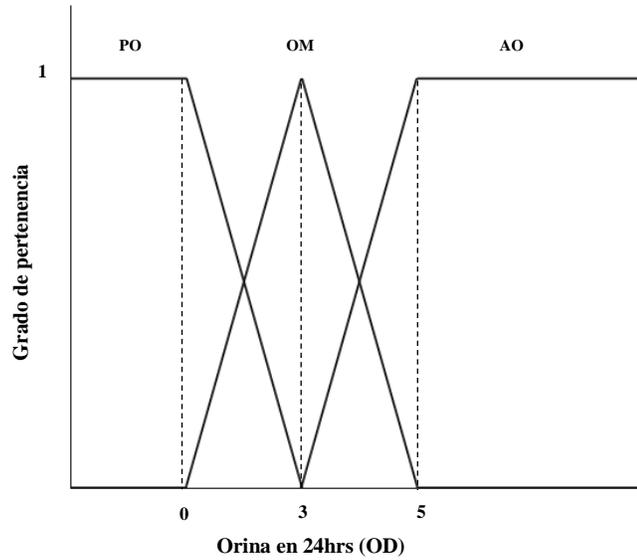


Figura 34 Conjunto difuso para el número de veces que orina el usuario en 24 horas.

VARIABLES LINGÜÍSTICAS:

PO: Nivel bajo de orina. | **OM:** Nivel medio de orina. | **AO:** Nivel alto de orina.

El conjunto de entrada para el número de micciones por la noche $f(MN)$ se muestra en la Figura 35.

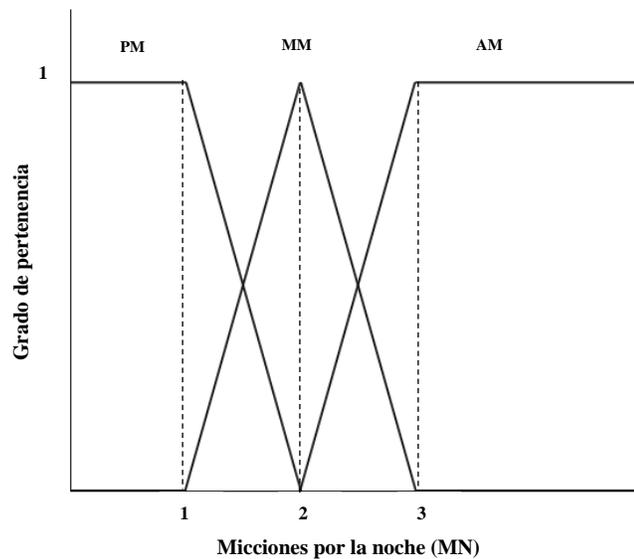


Figura 35 Conjunto difuso para el número de micciones por la noche.

VARIABLES LINGÜÍSTICAS:

PM: Nivel bajo de mixiones.	MM: Nivel medio de mixiones.	AM: Nivel alto de mixiones.
------------------------------------	-------------------------------------	------------------------------------

El conjunto de entrada para la presión arterial $f(PA)$ se muestra en la Figura 36.

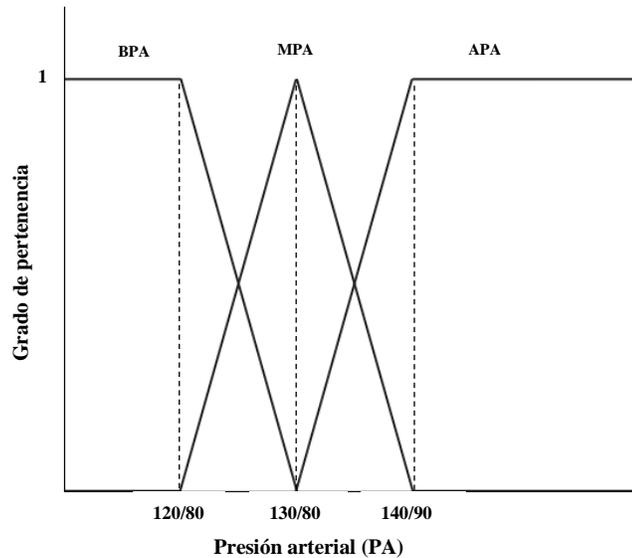


Figura 36 Conjunto difuso para el nivel de presión arterial.

VARIABLES LINGÜÍSTICAS:

BPA: Nivel bajo de presión arterial.	MPA: Nivel medio de presión arterial.	APA: Nivel alto de presión arterial.
---	--	---

Diseño del Fuzzificador

Se transformaron los valores de los conjuntos difusos para realizar el mecanismo de inferencia, esto permite manifestar el nivel de pertenencia de los conjuntos. Las funciones de inclusión para cada uno de los conjuntos de entrada se definen a continuación:

Para el conjunto de número de veces que orina en 24 horas se obtiene con la ecuación 12.

$$f(OD) = \begin{cases} PO = \begin{cases} PO = 0 & VO > 3 \\ PO = \frac{3 - VO}{3 - 0} & 1 \geq VO < 3 \end{cases} \\ OM = \begin{cases} OM = 0 & VO < 3 \\ OM = \frac{VO - 0}{3 - 0} & 0 > VO \leq 3 \\ OM = \frac{5 - VO}{5 - 3} & 3 > VO < 5 \end{cases} \\ AO = \begin{cases} AO = 0 & VO < 5 \\ AO = \frac{VO - 3}{5 - 3} & 3 > VO < 5 \\ AO = 1 & VO \geq 5 \end{cases} \end{cases} \quad (12)$$

Donde VO representa las veces que el usuario orina en un total de 24 horas.

El conjunto de micciones por la noche se obtiene con la ecuación 13.

$$f(MN) = \begin{cases} PM = \begin{cases} PM = 0 & M > 1 \\ PM = \frac{2 - M}{2 - 1} & M = 1 \end{cases} \\ MM = \begin{cases} MM = 0 & M < 2 \\ MM = \frac{M - 0}{2 - 0} & M = 2 \\ MM = \frac{3 - M}{3 - 2} & M > 2 \end{cases} \\ AM = \begin{cases} AO = 0 & M < 3 \\ AO = \frac{VO - 2}{3 - 2} & 2 > M < 3 \\ AO = 1 & M \geq 3 \end{cases} \end{cases} \quad (13)$$

Donde M es el número de micciones por la noche.

El conjunto de entrada para el nivel de presión arterial se determina por la ecuación 14.

$$f(PA) = \begin{cases} BPA = \begin{cases} BPA = 0 & P < 120/80 \\ BPA = \frac{130/80 - P}{130/80 - 120/80} & 120/80 \geq P < 130/80 \end{cases} \\ MPA = \begin{cases} MPA = 0 & P < 130/80 \\ MPA = \frac{P - 120/80}{130/80 - 120/80} & 120/80 > P < 130/80 \\ MPA = \frac{140/90 - P}{140/90 - 130/80} & 130/80 > P < 140/90 \end{cases} \\ APA = \begin{cases} APA = 0 & P < 140/90 \\ APA = \frac{P - 130/80}{140/90 - 130/80} & 130/80 > P < 140/90 \\ APA = 1 & P \geq 140/90 \end{cases} \end{cases} \quad (14)$$

Donde P es el nivel de presión arterial.

Diseño de reglas difusas

Las reglas difusas propuestas por Mamdani fueron utilizadas para el desarrollo del algoritmo, por ello, se realizaron las combinaciones entre cada uno de los conjuntos de entrada que se obtuvieron en la escala de Likert, el proceso se realizó mediante la premisa/consecuente.

Para este proyecto se implementó la regla Max-min mostrada en la ecuación 10, la cual consiste en buscar los valores mínimos de las reglas, posterior a ellos son comparadas una vez más para obtener el valor máximo, esto permitirá obtener los valores representativos para las etiquetas alto, medio y bajo.

Los datos para hacer el prediagnóstico se tomaron de la aplicación de la escala de tipo Likert, la cual abastecerá de una respuesta a cada uno de los ítems del cuestionario, para que así puedan generarse los conjuntos de entrada. Los ítems fueron determinados por un experto en el área de Nefrología, lo cual permite que los datos sean evaluados de tal manera que se aproximen al diagnóstico real, sin embargo, el objetivo del proyecto es únicamente apoyar al diagnóstico de la población y no sustituir al personal médico.

Tabla 5 Variables del modelo difuso.

Variable de entrada	Etiqueta lingüística	Función de pertenencia	Intervalos
Orina en 24hrs	Bajo	Trapezoidal	(0,3)
	Medio	Triangular	(0,3,5)
	Alto	Trapezoidal	(3,5)
Mixiones por la noche	Normal	Trapezoidal	(0,1,2)
	Medio	Triangular	(1,2,3)
	Mucho	Trapezoidal	(2,3)
Presión arterial	Estable	Trapezoidal	(0,120/80, 130/80)
	Regular	Triangular	(120/80, 130/80, 140/90)
	Elevada	Trapezoidal	(130/80, 140/90)
Diabetes	Si	Singleton	1
	No	Singleton	0
Medicamentos frecuentes	Aspirina	Singleton	1
	Aines	Singleton	2
	Antibiótico	Singleton	3
Ácido úrico	Si	Singleton	1
	No	Singleton	0
Enfermedades reumáticas	Si	Singleton	1
	No	Singleton	0
Familia con enfermedades renales	Si	Singleton	1
	No	Singleton	0
Familia con quistes renales	Si	Singleton	1
	No	Singleton	0
Ultrasonidos en 6 meses	Si	Singleton	1
	No	Singleton	0
Variable de salida	Etiqueta lingüística	Función de pertenencia	Intervalos
Enfermedad renal	Riesgo bajo	Trapezoidal	(0,36.36,50)
	Riesgo medio	Triangular	(36.36,50,100)
	Riesgo alto	Trapezoidal	(50,100)

En la Tabla 5, se muestran las variables que fueron utilizadas para la generación de los conjuntos difusos, las cuales permitirán generar el porcentaje de riesgo de padecer una enfermedad renal, se determinan tanto las variables de entrada como las de salida.

Cada variable permitió el diseño de un conjunto difuso, esto permite que se realice la evaluación de cada uno de ellos y determine el grado de pertenencia de las respuestas

aportadas por el usuario. Una vez obtenida una respuesta unificada, se realiza la defuzificación y da como resultado el porcentaje de riesgo.

Al término de la codificación del algoritmo, se realizaron pruebas del funcionamiento, esto para determinar el grado de veracidad que se tiene en la respuesta que genera el módulo de inferencias.

6 RESULTADOS Y DISCUSIÓN

En esta sección se presentan las pruebas a las cuales fue sometido el sistema, dichas pruebas tienen como propósito demostrar el buen funcionamiento de la aplicación móvil en cada una de sus vistas, esto permite que se detecten errores que se pudieron generar durante el desarrollo del sistema y se busque la eliminación de estos.

Interfaz de usuario

Como primera instancia, al momento de abrir la aplicación, aparece el logotipo de la aplicación por dos segundos, posterior a ello, aparece una introducción (Figura 37) a la aplicación, la cual consiste en una vista en la que se da una breve explicación de lo que contiene la aplicación, en la vista siguiente se da un aviso en el que se menciona que la aplicación no sustituirá al médico experto y en la vista final se da la bienvenida a los usuarios.

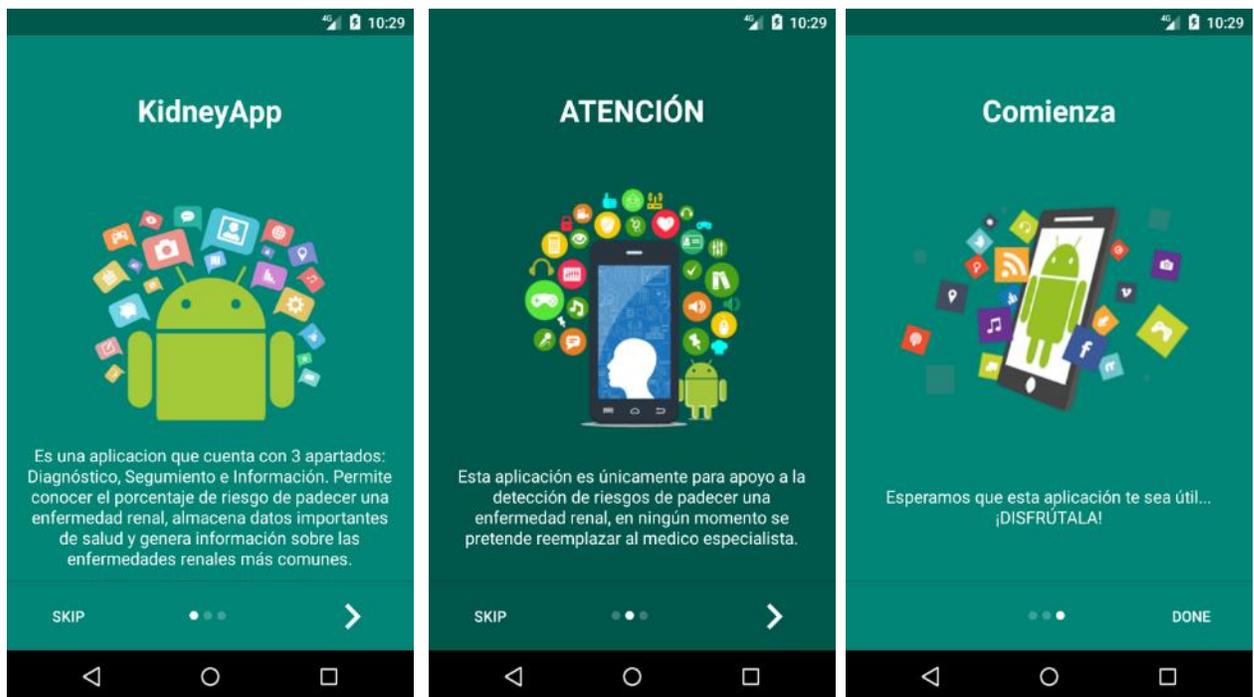


Figura 37 Introducción a la aplicación.

En la pantalla de inicio de sesión se colocó el logotipo característico de la aplicación, así como dos campos de texto, en los cuales se debe ingresar el nombre de usuario y la contraseña de acceso que el usuario determine para ser utilizada en la aplicación, así como un botón que permite determinar si el usuario tiene acceso al sistema o no mediante un registro previo que será almacenado en una base de datos. En caso de no estar registrado, se tiene un texto denominado **Registrar**, el cual al ser presionado se dirige a la pantalla de registro de usuario. Esta pantalla se puede observar en la Figura 38.



Figura 38 Pantalla de inicio de sesión.

En la pantalla de registro se muestra nuevamente el logotipo de la aplicación, así como tres campos de texto en los que se debe ingresar el nombre de usuario, la contraseña y una confirmación de esta, tal como se muestra en la Figura 39, al ser presionado el botón register, se almacena dicha información en la base de datos SQLite que se crea automáticamente, una vez registrado, se redirige a la pantalla principal para iniciar sesión, esto le permite tener una cuenta activa para tener acceso a cada una de las funciones con las que el sistema cuenta.



Figura 39 Pantalla de registro de usuario.

Una vez iniciada la sesión, la primera vista a la que tendrá acceso el usuario es al menú emergente, el cual, al dar clic sobre cualquiera de las imágenes, se redirige a la pantalla correspondiente, como se puede ver en la Figura 40, contiene cuatro opciones disponibles para que el usuario pueda interactuar con el sistema, las cuales son: diagnóstico, seguimiento, información y ajustes.



Figura 40 Menú de la aplicación.

Si se selecciona en el menú la opción de seguimiento, se muestra la pantalla de la Figura 41, se dirigirá a la vista correspondiente, en ella se tienen seis campos de texto en los que se deben ingresar los datos correspondientes a la etiqueta que tiene cada uno de ellos, esto para que sean almacenados en una nueva tabla de la base de datos y así se lleve el seguimiento del usuario.



Figura 41 Pantalla de seguimiento del paciente.

En caso de seleccionar la opción de información (Figura 42), se redirige a dicha vista y se muestra una secuencia de pantallas, las cuales irán cambiando cuando el usuario deslice el dedo por la pantalla, cada una de ellas proporcionan datos importantes sobre las diferentes enfermedades renales que actualmente existen, así como los síntomas más comunes que se pueden presentar.



Figura 42 Pantalla de información.

En la Figura 43 se muestra una pantalla que contiene un botón, el cual, al ser presionado, despliega un calendario en el que selecciona la fecha en la que tiene una próxima cita, una vez seleccionada la fecha se presiona el botón *Ok* y se muestra la fecha seleccionada en la pantalla para que se tenga presente. También tiene una imagen que permite salir de la aplicación una vez que se presiona.

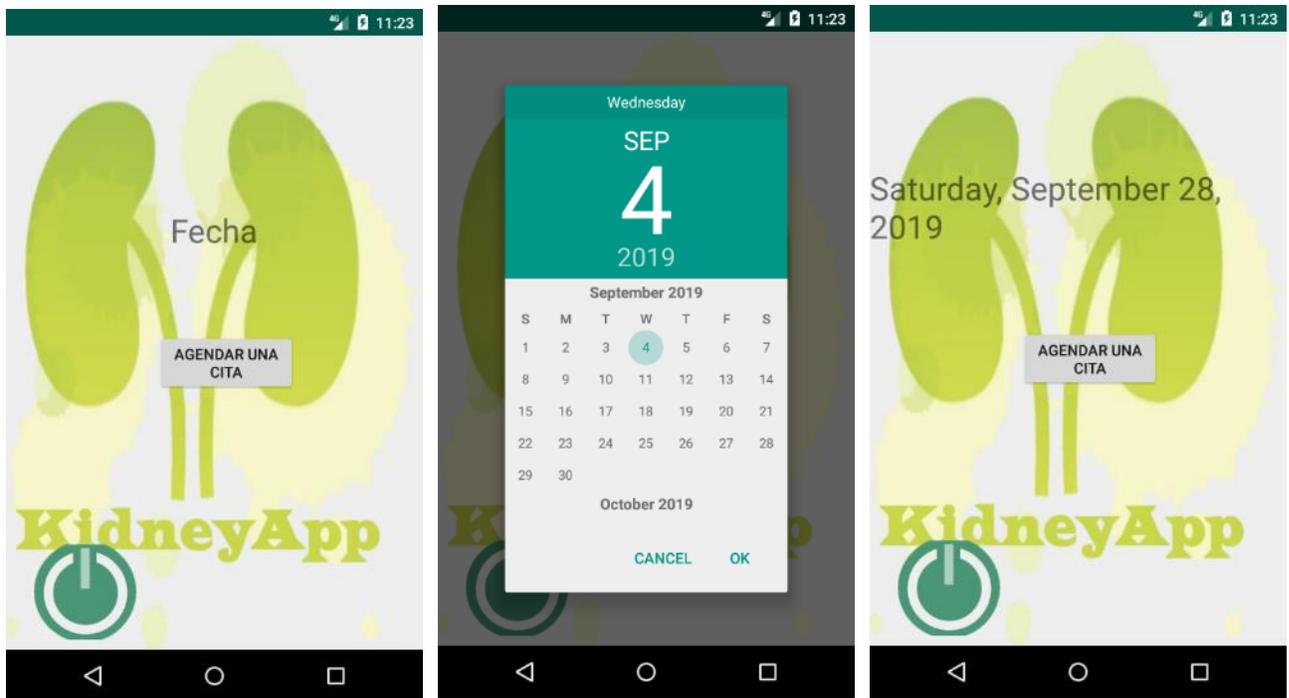


Figura 43 Agendar cita.

Para el caso del diagnóstico, se muestra una vista que contiene diez preguntas que deben ser respondidas por el usuario, se toma cada una de las respuestas y se diseña la función de pertenencia que para el caso de este modelo se hizo uso de funciones triangulares, trapezoidales y Singleton (Tabla 5) respectivamente, una vez construidos los conjuntos, fue necesario definir las ecuaciones 12 a 14, las cuales representan a los conjuntos de entrada definidos al comienzo del sistema. Fue necesario verificar cada una de las reglas difusas ya que no todos los ítems tenían el mismo número de variables lingüísticas.

Previo a la codificación del algoritmo, se realizaron pruebas manuales con diversos valores de entrada para verificar que, al momento de programarlo, los resultados serían los mismos, de no ser así, analizar el proceso y detectar el error. Posterior a realizar cada una de las etapas de la lógica difusa, se comenzó la codificación, esto se tornó más sencillo debido a que ya se conocía todo el proceso. En la Figura 44 se muestra la pantalla con la que interactúan los usuarios.

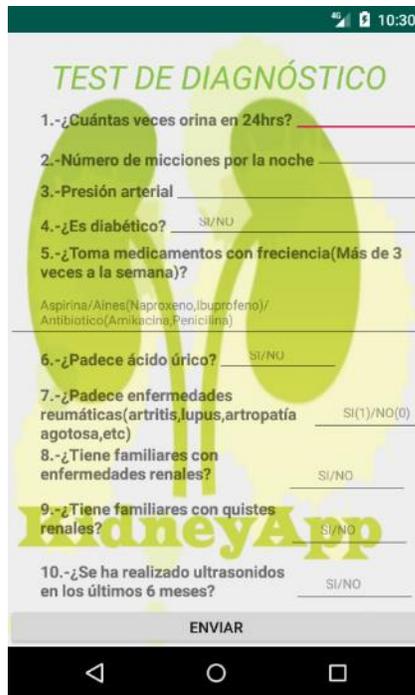


Figura 44 Pantalla para el diagnóstico.

Para verificar el funcionamiento de la aplicación, se realizaron pruebas en donde se contestaron las preguntas, algunos de los resultados se muestran en las tablas que a continuación se presentan, una vez escritas las respuestas de los usuarios en cada casilla, se da clic en el botón de Enviar para que el algoritmo se ejecute y realice la evaluación de casa una de las variables de entrada, este botón permite realizar cada una de las etapas de la lógica difusa, permitiendo así la generación del porcentaje de riesgo de cada uno de los usuarios.

Los resultados obtenidos de la prueba realizada a tres personas aleatorias se encuentran en las siguientes tablas.

Tabla 6 Prueba al usuario 1.

Ítem	Valores de entrada	Porcentaje asignado por el sistema
1.- ¿Cuántas veces orina en 24hrs?	2	82.05%
2.- Número de micciones por la noche.	1	
3.- Presión arterial	120	
4.- ¿Es diabético?	No	
5.- ¿Toma medicamentos con frecuencia (Más de 3 veces a la semana)?	Aspirina	
6.- ¿Padece ácido úrico?	No	
7.- ¿Padece enfermedades reumáticas (artritis, lupus, artropatía agotosa, etc)?	No	
8.- ¿Tiene familiares con enfermedades renales?	Si	
9.- ¿Tiene familiares con quistes renales?	No	
10.- ¿Se ha realizado ultrasonidos en los últimos 6 meses?	No	

Tabla 7 Prueba al usuario 2.

Ítem	Valores de entrada	Porcentaje asignado por el sistema
1.- ¿Cuántas veces orina en 24hrs?	5	100.00%
2.- Número de micciones por la noche.	3	
3.- Presión arterial	140	
4.- ¿Es diabético?	Si	
5.- ¿Toma medicamentos con frecuencia (Más de 3 veces a la semana)?	Antibióticos	
6.- ¿Padece ácido úrico?	Si	
7.- ¿Padece enfermedades reumáticas (artritis, lupus, artropatía agotosa, etc)?	Si	
8.- ¿Tiene familiares con enfermedades renales?	Si	
9.- ¿Tiene familiares con quistes renales?	Si	
10.- ¿Se ha realizado ultrasonidos en los últimos 6 meses?	No	

Tabla 8 Prueba al usuario 3.

Ítem	Valores de entrada	Porcentaje asignado por el sistema
1.- ¿Cuántas veces orina en 24hrs?	3	54.54%
2.- Número de micciones por la noche.	1	
3.- Presión arterial	130	
4.- ¿Es diabético?	Si	
5.- ¿Toma medicamentos con frecuencia (Más de 3 veces a la semana)?	No	
6.- ¿Padece ácido úrico?	No	
7.- ¿Padece enfermedades reumáticas (artritis, lupus, artropatía agotosa, etc)?	Si	
8.- ¿Tiene familiares con enfermedades renales?	No	
9.- ¿Tiene familiares con quistes renales?	No	
10.- ¿Se ha realizado ultrasonidos en los últimos 6 meses?	Si	

Tabla 9 Prueba al usuario 4.

Ítem	Valores de entrada	Porcentaje asignado por el sistema
1.- ¿Cuántas veces orina en 24hrs?	1	79.72%
2.- Número de micciones por la noche.	0	
3.- Presión arterial	90	
4.- ¿Es diabético?	No	
5.- ¿Toma medicamentos con frecuencia (Más de 3 veces a la semana)?	Aines	
6.- ¿Padece ácido úrico?	No	
7.- ¿Padece enfermedades reumáticas (artritis, lupus, artropatía agotosa, etc)?	Si	
8.- ¿Tiene familiares con enfermedades renales?	No	
9.- ¿Tiene familiares con quistes renales?	Si	
10.- ¿Se ha realizado ultrasonidos en los últimos 6 meses?	No	

Tabla 10 Prueba al usuario 5.

Ítem	Valores de entrada	Porcentaje asignado por el sistema
1.- ¿Cuántas veces orina en 24hrs?	9	54.54%
2.- Número de micciones por la noche.	0	
3.- Presión arterial	180	
4.- ¿Es diabético?	Si	
5.- ¿Toma medicamentos con frecuencia (Más de 3 veces a la semana)?	No	
6.- ¿Padece ácido úrico?	No	
7.- ¿Padece enfermedades reumáticas (artritis, lupus, artropatía agotosa, etc)?	No	
8.- ¿Tiene familiares con enfermedades renales?	No	
9.- ¿Tiene familiares con quistes renales?	No	
10.- ¿Se ha realizado ultrasonidos en los últimos 6 meses?	No	

Tabla 11 Prueba al usuario 6.

Ítem	Valores de entrada	Porcentaje asignado por el sistema
1.- ¿Cuántas veces orina en 24hrs?	2	86.24%
2.- Número de micciones por la noche.	1	
3.- Presión arterial	70	
4.- ¿Es diabético?	No	
5.- ¿Toma medicamentos con frecuencia (Más de 3 veces a la semana)?	Aspirina	
6.- ¿Padece ácido úrico?	Si	
7.- ¿Padece enfermedades reumáticas (artritis, lupus, artropatía agotosa, etc)?	No	
8.- ¿Tiene familiares con enfermedades renales?	No	
9.- ¿Tiene familiares con quistes renales?	No	
10.- ¿Se ha realizado ultrasonidos en los últimos 6 meses?	No	

Estas pruebas permitieron conocer el porcentaje de riesgo que tienen estas personas de padecer una enfermedad renal, así como verificar que la aplicación estaba funcionando

correctamente, así como automatizar conocimiento empírico por parte de médicos expertos, esto para que esté al alcance de más personas y les permita así estar más informados acerca de este tipo de enfermedades y tomen precauciones de ser necesario.

CONCLUSIONES

El principal propósito del proyecto que se presenta en este documento es desarrollar una aplicación móvil que permita simular la evaluación médica de pacientes con padecimientos renales, sin necesidad de usar datos estrictamente clínicos, como lo son los niveles de creatinina sérica, filtrado glomerular, albuminuria-proteinuria, sedimento urinario, estudios de imagen o histopatológicos, los cuales no se pueden conocer sin asistir a un laboratorio médico. El sistema experto, diseñado para el diagnóstico y seguimiento de enfermedades renales, permite generar un diagnóstico previo, el cual pretende concientizar a los usuarios acerca del grado de riesgo en el que se encuentran, y de esta forma asistan con un especialista para que la enfermedad pueda ser detectada a tiempo y brindar el tratamiento adecuado.

En diversas aplicaciones, el uso de la lógica difusa ha permitido la automatización de procesos basados en el conocimiento o la experiencia humana. De manera similar, en este proyecto, la impresión de los datos manejados en el cuestionario permite que la lógica difusa se adecúe para brindar una respuesta semejante a la que daría un experto en el área de nefrología, expresando mediante una valoración en porcentaje de riesgo, la posibilidad de que exista un padecimiento de este tipo en la persona, a partir de respuestas a preguntas simples sobre su vida cotidiana y antecedentes familiares.

El análisis de cada uno de los ítems utilizados puede sugerir a los usuarios, dependiendo del porcentaje asignado, que consulten a un médico para que realice el diagnóstico oficial y determine el proceso a seguir según su estadio, así como informarle de las enfermedades existentes y los síntomas para que se mantenga al tanto de algún cambio que se presente en su organismo y puedan acudir al médico para una evaluación más profunda.

Cabe mencionar que suplantar el conocimiento de un médico por el uso de esta aplicación sería una equivocación, ya que únicamente funciona como una herramienta de apoyo que, por su facilidad de uso y compatibilidad con un celular, tiene el potencial de ser utilizado por usuarios diversos. Además, puede ayudar a la población a tener más información acerca de estos padecimientos y pongan mayor atención, ya que las enfermedades renales son una de las principales causantes de muerte en el mundo y actualmente no se ha brindado la suficiente información.

Es importante destacar que para este proyecto se hizo uso de la metodología en V para el desarrollo de software, ya que permite una implementación ágil de la aplicación móvil. Al mismo tiempo, se utilizaron diferentes herramientas relacionadas con la Ingeniería en Computación, tales como el desarrollo de una base de datos, programación en Java mediante el entorno de Android Studio, Diseño Orientado a Objetos, programación de interfaces de usuario y Sistemas de Inferencia Difusos.

Como trabajos a futuro, se proponen las siguientes acciones:

- Optimizar el módulo de inferencias que se manejan para tener una mayor aproximación al diagnóstico de enfermedades renales.
- Ampliar las pruebas de funcionamiento, así como realizar comparaciones con el diagnóstico médico y los resultados proporcionados por el sistema.
- Ampliar las reglas difusas para tener un diagnóstico más amplio.
- Implementar diferentes reglas de inclusión para los conjuntos de entrada.

REFERENCIAS

Alonso Arévalo, J., 2016. *Aplicaciones móviles en medicina y salud*. [En línea]

Available at:

<https://gredos.usal.es/jspui/bitstream/10366/130118/1/Aplicaciones%20m%C3%B3viles%20en%20medicina%20y%20salud.pdf>

[Último acceso: 24 septiembre 2018].

Alonso Arévalo, J. & Mirón Canelo, J. A., 2017. Aplicaciones móviles en salud: potencial, normativa de seguridad y regulación. *Revista Cubana de Información en Ciencias de la Salud*, 22 septiembre, Volumen 3, pp. 1-13.

Ampuero, J. C., 2008. Redes neuronales de base radial aplicadas a la mejora de la calidad. *Revista de la Facultad de Ingeniería Industrial*, pp. 63-72.

Anaya Fernandez, S., Vozmediano Poyatos, C. & Rivera Hernández, F., 2012. Síndromes clínicos en nefrología. *Nefrología al día*, Volumen 1, pp. 3-22.

Barrio, V., 2012. Necesidad y utilidad del empleo de criterios estandarizados para el diagnóstico de la disfunción renal aguda en pacientes críticos. *Medicina intensiva*, 36(4), pp. 247-249.

Barrio, V., 2012. Necesidad y utilidad del empleo de criterios estandarizados para el diagnóstico de la disfunción renal aguda en pacientes críticos. *Medicina intensiva*, 36(4), pp. 247-249.

Beltrán Ramírez, R., Maciel Arellano, R. & Jiménez Arévalo, J., 2014. La tecnología y la inteligencia artificial como futuro en el área médica. *Universitas Revista de ciencias sociales y humanas*, 15 diciembre, Volumen 21, pp. 185-190.

Bonilla, D., 2012. Algoritmos evolutivos en el diagnóstico de tumores de mama. *Seminario de ingeniería biomédica*, pp. 1-5.

Bracamonte Alaniz, X., 2008. La Inteligencia Artificial en la Medicina. *Revista de Información, Tecnología y Sociedad*, Volumen 1, pp. 38-39.

Cárdenas, P. D. A. G., 2011. *Cómo leer un artículo científico*, Colombia: Departamento de Ciencias Biológicas.

Clínica Renalis, 2016. *Renalis*. [En línea]

Available at: <https://www.renalis.com.mx/la-insuficiencia-renal-en-mexico/>

[Último acceso: 04 Julio 2019].

CNN, 2015. Las enfermedades que más afectan la vida de los mexicanos. *EXPANSIÓN*, 26 08, pp. 1-3.

Cuma, C. C., 2017. *Metodología de Desarrollo de Software - ULADECH*. [En línea]

Available at:

<https://www.uladech.edu.pe/images/stories/universidad/documentos/2018/metodologia-desarrollo-software-v001.pdf>

[Último acceso: 15 enero 2019].

de Paula Cerqueira, D., Tavares, J. R. & Machado, C. R., 2014. Factores de predicción de la insuficiencia renal y el algoritmo de control y tratamiento. *Revista Latino-Americana de Enfermagem*, abril, Volumen 2, pp. 211-217.

Delsol, K., Isaia, M. & Minetti, G., 2015. Aplicación de algoritmos aproximados al diagnóstico/clasificación de enfermedades. *Simposio Argentino de Inteligencia Artificial*, 16(2451-7585), pp. 49-56.

EcuRed, 2019. *EcuRed*. [En línea]

Available at: <https://www.ecured.cu/Terap%C3%A9utica>

[Último acceso: 07 07 2019].

Fundación clínica para la reseva biomédica, 2015. *BioTech*. [En línea]

Available at: <http://biotech-spain.com/es/articles/nuevo-algoritmo-para-una-nueva-definicion-de-la-insuficiencia-renal-aguda-en-pacientes-con-cirrosis/>

[Último acceso: 1 noviembre 2018].

Gorostidi, M. y otros, 2014. Documento de la sociedad española de nefrología sobre las guías KDIGO para la evaluación y el tratamiento de la enfermedad renal crónica.

Revista nefrología. Órgano oficial de la sociedad española de nefrología, 34(10.3265), pp. 302-316.

Grau, I. y otros, 2016. Método de valoración de aplicaciones móviles de salud en español: el índice iSYScore. *Medicina de Familia SEMERGEN*, 42(8), pp. 575-583.

Instituto Mexicano del Seguro Social, 2019. Prevención, diagnóstico y tratamiento de la enfermedad renal crónica temprana. *Guía de referencia rápida*, Volumen 1, pp. 1-20.

Lugo Reyes, S. O., Maldonado Colín, G. & Murata, C., 2014. Inteligencia artificial para asistir el diagnóstico clínico en medicina. *Revista Alegria México*, marzo, Volumen 61, pp. 110-120.

M., Y. M., 2018. *OpenWebinars*. [En línea]

Available at: <https://openwebinars.net/blog/sqlite-para-android-la-herramienta-definitiva/>

[Último acceso: 14 enero 2019].

Maese Núñez, J. d. D., Alvarado Iniesta, A., Valles Rosales, D. J. & Báez López, Y. A., 2016. Coeficiente alfa de Cronbach para medir la fiabilidad de un cuestionario difuso. *Culcyt/Instrumentos de medición*, 21 08, pp. 146-156.

Martinez Alvarado, R. y otros, 2018. Red neuronal con función de base radial para modelar un proceso de perforación mecanizado por descarga eléctrica. *Reunión Internacional de Otoño IEEE 2018 sobre Energía, Electrónica e Informática (ROPEC 2018)*, pp. 1-6.

Martínez Castelo, A. y otros, 2014. Documento de consenso para la detección y manejo de la enfermedad renal crónica. *Revista nefrología. Órgano oficial de la sociedad española de nefrología*, 34(0212-6567), pp. 243-62.

Morais Arruda Costa, G. y otros, 2016. Calidad de vida en pacientes con insuficiencia renal crónica en hemodiálisis. *Enfermería Global*, julio, Volumen 15, pp. 59-72.

Morcillo, C. G., 2019. *Lógica difusa una introducción práctica*. [En línea]

Available at:

https://www.esi.uclm.es/www/cglez/downloads/docencia/2011_Softcomputing/LogicaDifusa.pdf

[Último acceso: 21 08 2019].

Pascual, J. A., 2018. *Computer Hoy*. [En línea]

Available at: <https://computerhoy.com/reportajes/industria/android-vs-iphone-guerra-smartphones-cifras-271447>

[Último acceso: 04 Julio 2019].

Perez, A. y otros, 2006. Una metodología para el desarrollo de hardware y software embebido en sistemas críticos de seguridad. *Sistemas, cibernética e informática*, 3(1690-8627), pp. 70-75.

Plan Seguro, 2019. *PLAN SEGURO*. [En línea]

Available at: <https://blog.planseguro.com.mx/el-uso-de-las-app-en-la-industria-de-la-salud>

[Último acceso: 07 07 2019].

Raff, J., 2013. *Cómo leer y entender un artículo científico: una guía para no científicos*.

[En línea]

Available at: <https://violentmetaphors.com/2013/08/25/how-to-read-and-understand-a-scientific-paper-2/>

Redacción Tecnósfera, 2017. *Inteligencia artificial aplicada a la medicina*. [En línea]

Available at: <https://www.eltiempo.com/tecnosfera/novedades-tecnologia/como-funciona-la-inteligencia-artificial-en-la-medicina-101730>

[Último acceso: 08 septiembre 2018].

Ruben, A., 2016. *Science*. [En línea]

Available at: <http://www.sciencemag.org/careers/2016/01/how-read-scientific-paper>

Sáa, S. d. I. R. d., 2012. Análisis estadístico comparativo de tres escalas de valoración: Likert, fuzzy-Likert y fuzzy de respuesta libre. *Máster en Modelización Matemática, Estadística y Computación*, 19 julio, pp. 1-90.

SQLite, 2019. *SQLite*. [En línea]

Available at: <https://www.sqlite.org/about.html>

[Último acceso: 29 08 2019].

Stallings, W., 2006. *Cryptography and network security Principles and practice*. 5ta ed. Estados Unidos de América: 2011.

Velasco Rodríguez, G., 2018. Mercadotecnia social: las aplicaciones móviles en el mercado sanitario. *Horizonte sanitario*, enero, Volumen 17, pp. 9-20.

Vergottini, J. C. y otros, 2013. Detección de riesgo para enfermedad renal en adolescentes. *Revista de Salud Pública*, julio, Volumen 17, pp. 63-72.

Villamizar Suaza, K., Soto Durán, D. E., Giraldo Mejía, J. C. & Jiménez Builes, J. A., 2016. Modelo de pruebas en proyectos BI. *Engineering Innovations for Global Sustainability*, 14(2414-6390), pp. 1-9.

ANEXOS

SplashActivity

```
package com.aletvia.appintro;  
  
import android.content.Intent;  
import android.content.pm.ActivityInfo;  
import android.os.Handler;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.WindowManager;  
  
public class SplashActivity extends AppCompatActivity {  
  
    private final int DURACION_SPLASH = 2000;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);  
        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
WindowManager.LayoutParams.FLAG_FULLSCREEN);  
  
        setContentView(R.layout.activity_splash);  
  
        new Handler().postDelayed(new Runnable(){  
            public void run(){  
                Intent intent = new Intent(SplashActivity.this, AppIntroActivity.class);  
                startActivity(intent);  
                finish();  
            }  
        }, DURACION_SPLASH);  
    }  
}
```

AppIntroActivity

```
package com.aletvia.appintro;  
  
import android.content.Context;  
import android.content.Intent;  
import android.support.v4.app.Fragment;  
import android.support.v4.content.ContextCompat;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
  
import com.github.paolorotolo.appintro.AppIntro;  
import com.github.paolorotolo.appintro.AppIntro2Fragment;  
import com.github.paolorotolo.appintro.AppIntroFragment;  
  
public class AppIntroActivity extends AppIntro {
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    //setContentView(R.layout.activity_app_intro);

    addSlide(AppIntroFragment.newInstance("KidneyApp", "Es una aplicación que cuenta con 3
apartados: Diagnóstico, Seguimiento e Información. Permite conocer el porcentaje de riesgo de
padecer una enfermedad renal, almacena datos importantes de salud y genera información sobre
las enfermedades renales más comunes.",
        R.drawable.one, ContextCompat.getColor(getApplicationContext(),R.color.colorPrimary));
    addSlide(AppIntroFragment.newInstance("ATENCIÓN", "Esta aplicación es únicamente para
apoyo a la detección de riesgos de padecer una enfermedad renal, en ningún momento se
pretende reemplazar al medico especialista.", R.drawable.two,
ContextCompat.getColor(getApplicationContext(),
    R.color.colorPrimaryDark));
    addSlide(AppIntroFragment.newInstance("Comienza", "Esperamos que esta aplicación te sea
útil... ¡DISFRÚTALA!",
        R.drawable.three,
ContextCompat.getColor(getApplicationContext(),R.color.colorPrimary));
}

@Override
public void onDonePressed(Fragment currentFragment){
    super.onDonePressed(currentFragment);
    Intent intent = new Intent(getApplicationContext(), LoginActivity.class);
    startActivity(intent);
}

@Override
public void onSkipPressed(Fragment currentFragment){
    super.onSkipPressed(currentFragment);
    Intent intent = new Intent(getApplicationContext(), LoginActivity.class);
    startActivity(intent);
}
}

```

LoginActivity

```

package com.aletvia.appintro;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class LoginActivity extends AppCompatActivity {

    EditText mTextUsername;
    EditText mTextPassword;
}

```

```

Button mButtonLogin;
TextView mTextViewRegister;
DatabaseHelper db;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);

    db = new DatabaseHelper(this);
    mTextUsername = (EditText)findViewById(R.id.edittext_username);
    mTextPassword = (EditText)findViewById(R.id.edittext_password);
    mButtonLogin = (Button) findViewById(R.id.button_login);
    mTextViewRegister = (TextView) findViewById(R.id.textview_register);
    mTextViewRegister.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent registerIntent = new Intent(LoginActivity.this, RegisterActivity.class);
            startActivity(registerIntent);
        }
    });

    mButtonLogin.setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick(View view){
            String user = mTextUsername.getText().toString().trim();
            String pwd = mTextPassword.getText().toString().trim();
            Boolean res = db.CheckUser(user, pwd);
            if (res == true){
                Toast.makeText(LoginActivity.this, "Successfully logged In",
                Toast.LENGTH_SHORT).show();
                Intent Homepage = new Intent(LoginActivity.this, MenuActivity.class);
                startActivity(Hompage);
            }
            else {
                Toast.makeText(LoginActivity.this, "Login Error", Toast.LENGTH_SHORT).show();
            }
        }
    });
}
}

```

DatabaseHelper

```

package com.aletvia.appintro;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DatabaseHelper extends SQLiteOpenHelper {

```

```

public static final String DATABASE_NAME = "register.db";
public static final String TABLE_NAME = "registeruser";
public static final String COL_1 = "ID";
public static final String COL_2 = "username";
public static final String COL_3 = "password";

public DatabaseHelper(Context context) {
    super(context, DATABASE_NAME, null, 1);
}

@Override
public void onCreate(SQLiteDatabase sqLiteDatabase) {
    sqLiteDatabase.execSQL("CREATE TABLE registeruser (ID INTEGER PRIMARY KEY
    AUTOINCREMENT, username TEXT, password TEXT)");
}

@Override
public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
    sqLiteDatabase.execSQL(" DROP TABLE IF EXISTS " + TABLE_NAME);
    onCreate(sqLiteDatabase);
}

public long addUser(String user, String password){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put("username", user);
    contentValues.put("password", password);
    long res = db.insert("registeruser", null, contentValues);
    db.close();
    return res;
}

public boolean CheckUser(String username, String password){
    String[] columns = {COL_1};
    SQLiteDatabase db = getReadableDatabase();
    String selection = COL_2 + "=? " + " and " + COL_3 + "=?";
    String[] selectionArgs = {username, password};
    Cursor cursor = db.query(TABLE_NAME, columns, selection, selectionArgs, null,null,null);
    int count = cursor.getCount();
    db.close();

    if (count > 0)
        return true;
    else
        return false;
}
}

```

RegisterActivity

```

package com.aletvia.appintro;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;

```

```

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class RegisterActivity extends AppCompatActivity {

    DatabaseHelper db;
    EditText mTextUsername;
    EditText mTextPassword;
    EditText mTextCnfPassword;
    Button mButtonRegister;
    TextView mTextViewLogin;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        db = new DatabaseHelper(this);
        mTextUsername = (EditText)findViewById(R.id.edittext_username);
        mTextPassword = (EditText)findViewById(R.id.edittext_password);
        mTextCnfPassword = (EditText)findViewById(R.id.edittext_cnf_password);
        mButtonRegister = (Button) findViewById(R.id.button_register);
        mTextViewLogin = (TextView) findViewById(R.id.textview_login);
        mTextViewLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent LoginIntent = new Intent(RegisterActivity.this, LoginActivity.class);
                startActivity(LoginIntent);
            }
        });

        mButtonRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String user = mTextUsername.getText().toString().trim();
                String pwd = mTextPassword.getText().toString().trim();
                String cnf_pwd = mTextCnfPassword.getText().toString().trim();

                if (pwd.equals(cnf_pwd)){
                    long val = db.addUser(user, pwd);
                    if (val > 0){
                        Toast.makeText(RegisterActivity.this, "You have registered",
                        Toast.LENGTH_SHORT).show();
                        Intent moveToLogin = new Intent(RegisterActivity.this, LoginActivity.class);
                        startActivity(moveToLogin);
                    }
                    else{
                        Toast.makeText(RegisterActivity.this, "Registration Error",
                        Toast.LENGTH_SHORT).show();
                    }
                }
            }
        });
    }
}

```

```

        else {
            Toast.makeText(RegisterActivity.this, "Password is not matching",
                Toast.LENGTH_SHORT).show();
        }
    }
});
}
}
}

```

MenuActivity

```

package com.aletvia.appintro;

import android.content.Intent;
import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Toast;
import com.hitomi.cmlibrary.CircleMenu;
import com.hitomi.cmlibrary.OnMenuSelectedListener;

public class MenuActivity extends AppCompatActivity {

    String arrayName[]={"Diagnostico",
        "Seguimiento",
        "Informacion",
        "Ajustes"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menu);

        CircleMenu circleMenu = (CircleMenu) findViewById(R.id.circle_menu);
        circleMenu.setMainMenu(Color.parseColor("#CDCDCD"), R.mipmap.icon_menu,
            R.mipmap.icon_cancel)
            .addSubMenu(Color.parseColor("#FF6A00"), R.mipmap.icon_diagnostico)
            .addSubMenu(Color.parseColor("#30A400"), R.mipmap.icon_seguimiento)
            .addSubMenu(Color.parseColor("#FF4B32"), R.mipmap.icon_informacion)
            .addSubMenu(Color.parseColor("#8A39FF"), R.mipmap.icon_ajustes)
            .setOnMenuSelectedListener(new OnMenuSelectedListener() {
                @Override
                public void onMenuSelected(int index) {
                    Toast.makeText(MenuActivity.this, "You selected " + arrayName[index],
                        Toast.LENGTH_SHORT).show();

                    if(arrayName[index] == "Informacion") {
                        Intent Homepage = new Intent(MenuActivity.this, IntroActivity.class);
                        startActivity(Homepage);
                    } else if(arrayName[index] == "Seguimiento") {
                        Intent Homepage = new Intent(MenuActivity.this, MainActivity.class);
                        startActivity(Homepage);
                    } else if (arrayName[index] == "Ajustes") {
                        Intent Homepage = new Intent(MenuActivity.this, FinalActivity.class);
                        startActivity(Homepage);
                    }
                }
            });
    }
}

```

```

        } else if (arrayName[index] == "Diagnostico"){
            Intent Homepage = new Intent(MenuActivity.this, Diagnostic.class);
            startActivity(Hompage);
        }
    }
});
}
}
}

```

MainActivity

```

package com.aletvia.appintro;

import android.content.ContentValues;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    private String[] header = {"Id", "Peso", "Estatura", "Medicamentos", "Alergias", "Tipo de
sangre"};
    private String shortText = "Hola";
    private String longText = "jbcjshbcjhsbdcjksbdcjhzsdbchjbjkj dasbjanbdjcnadb jasnbdcje,sa";
    Button btnGuardar, btnBuscar, btnActualizar, btnBorrar;
    EditText etId, etNombres, etTelefono, etEstatura, etMedicamentos, etAlergias;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnGuardar = (Button)findViewById(R.id.btnGuardar);
        btnBuscar = (Button)findViewById(R.id.btnBuscar);
        btnBorrar = (Button)findViewById(R.id.btnBorrar);
        btnActualizar = (Button)findViewById(R.id.btnActulizar);

        etId = (EditText)findViewById(R.id.etId);
        etNombres = (EditText)findViewById(R.id.etNombres);
        etEstatura = (EditText)findViewById(R.id.etEstatura);
        etMedicamentos = (EditText)findViewById(R.id.etMedicamentos);
        etAlergias = (EditText)findViewById(R.id.etAlergias);
        etTelefono = (EditText)findViewById(R.id.etTelefono);

        final AyudaBD ayudabd = new AyudaBD(getApplicationContext());

        btnGuardar.setOnClickListener(new View.OnClickListener() {

```

```

@Override
public void onClick(View v) {

    SQLiteDatabase db = ayudabd.getWritableDatabase();
    ContentValues valores = new ContentValues();
    valores.put(AyudaBD.DatosTabla.COLUMNNA_ID,etId.getText().toString());
    valores.put(AyudaBD.DatosTabla.COLUMNNA_NOMBRES,
etNombres.getText().toString());
    valores.put(AyudaBD.DatosTabla.COLUMNNA_ESTATURA,
etEstatura.getText().toString());
    valores.put(AyudaBD.DatosTabla.COLUMNNA_MEDICAMENTOS,
etMedicamentos.getText().toString());
    valores.put(AyudaBD.DatosTabla.COLUMNNA_ALERGIAS,
etAlergias.getText().toString());
    valores.put(AyudaBD.DatosTabla.COLUMNNA_TELEFONOS,
etTelefono.getText().toString());

    Long IdGuardado = db.insert(AyudaBD.DatosTabla.NOMBRE_TABLA,
AyudaBD.DatosTabla.COLUMNNA_ID, valores);
    Toast.makeText(getApplicationContext(), "Se guardo el dato: "+IdGuardado,
Toast.LENGTH_LONG).show();
}
});

btnBuscar.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
try {
        SQLiteDatabase db = ayudabd.getReadableDatabase();
        String[] argsel = {etId.getText().toString()};
        String[] projection = {AyudaBD.DatosTabla.COLUMNNA_NOMBRES,
AyudaBD.DatosTabla.COLUMNNA_ESTATURA,
AyudaBD.DatosTabla.COLUMNNA_MEDICAMENTOS,
AyudaBD.DatosTabla.COLUMNNA_ALERGIAS, AyudaBD.DatosTabla.COLUMNNA_TELEFONOS};
        Cursor c = db.query(AyudaBD.DatosTabla.NOMBRE_TABLA, projection,
AyudaBD.DatosTabla.COLUMNNA_ID+"=?",argsel,null,null,null);

        c.moveToFirst();
        etNombres.setText(c.getString(0));
        etEstatura.setText(c.getString(1));
        etMedicamentos.setText(c.getString(2));
        etAlergias.setText(c.getString(3));
        etTelefono.setText(c.getString(4));
    }catch (Exception e){
        Toast.makeText(getApplicationContext(), "No se encontró el registro",
Toast.LENGTH_LONG).show();
    }
}
});

btnBorrar.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {

    SQLiteDatabase db = ayudabd.getWritableDatabase();

```

```

String Selection = AyudaBD.DatosTabla.COLUMNA_ID+"=?";
String[] argsel = {etId.getText().toString()};

db.delete(AyudaBD.DatosTabla.NOMBRE_TABLA,Selection,argsel);
}
});

btnActualizar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SQLiteDatabase db = ayudabd.getWritableDatabase();
        ContentValues valores = new ContentValues();
        valores.put(AyudaBD.DatosTabla.COLUMNA_NOMBRES,
etNombres.getText().toString());
        valores.put(AyudaBD.DatosTabla.COLUMNA_ESTATURA,
etEstatura.getText().toString());
        valores.put(AyudaBD.DatosTabla.COLUMNA_MEDICAMENTOS,
etMedicamentos.getText().toString());
        valores.put(AyudaBD.DatosTabla.COLUMNA_ALERGIAS,
etAlergias.getText().toString());
        valores.put(AyudaBD.DatosTabla.COLUMNA_TELEFONOS,
etTelefono.getText().toString());
        String[] argsel = {etId.getText().toString()};
        String Selection = AyudaBD.DatosTabla.COLUMNA_ID+"=?";

        int count = db.update(AyudaBD.DatosTabla.NOMBRE_TABLA,valores,Selection,argsel);

    }
});

}

private ArrayList<String[]> getClients(){
    ArrayList<String[]>rows = new ArrayList<>();

    rows.add(new String[]{"1", "45", "160", "No consumo medicamentos", "No tengo alergias",
"A+"});
    return rows;
}
}
}

```

AyudaBD

```

package com.aletvia.appintro;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.provider.BaseColumns;

public class AyudaBD extends SQLiteOpenHelper {

    public static abstract class DatosTabla implements BaseColumns {

```

```

public static final String NOMBRE_TABLA = "Directorio";
public static final String COLUMNA_ID = "id";
public static final String COLUMNA_NOMBRES = "nombres";
public static final String COLUMNA_ESTATURA = "estatura";
public static final String COLUMNA_MEDICAMENTOS = "medicamentos";
public static final String COLUMNA_ALERGIAS = "alergias";
public static final String COLUMNA_TELEFONOS = "telefono";

private static final String TEXT_TYPE = "TEXT";
private static final String COMMA_SEP = ",";
private static final String CREAR_TABLA_I =
    "CREATE TABLE " + DatosTabla.NOMBRE_TABLA + " (" +
        DatosTabla.COLUMNA_ID + " INTEGER PRIMARY KEY," +
        DatosTabla.COLUMNA_NOMBRES + TEXT_TYPE + COMMA_SEP +
        DatosTabla.COLUMNA_ESTATURA + TEXT_TYPE + COMMA_SEP +
        DatosTabla.COLUMNA_MEDICAMENTOS + TEXT_TYPE + COMMA_SEP +
        DatosTabla.COLUMNA_ALERGIAS + TEXT_TYPE + COMMA_SEP +
        DatosTabla.COLUMNA_TELEFONOS + TEXT_TYPE +
    ")";

private static final String SQL_DELETE_ENTRIES =
    "DROP TABLE IF EXISTS " + DatosTabla.NOMBRE_TABLA;

}

public static final int DATABASE_VERSION = 1;
public static final String DATABASE_NAME = "MiBasedeDatos.db";

public AyudaBD(Context context) {

    super(context, DATABASE_NAME, null, DATABASE_VERSION);
}

@Override
public void onCreate(SQLiteDatabase db) {

    db.execSQL(DatosTabla.CREAR_TABLA_I);

}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    db.execSQL(DatosTabla.SQL_DELETE_ENTRIES);
    onCreate(db);

}
}

```

Diagnostic

```

package com.aletvia.appintro;

import android.content.DialogInterface;
import android.support.v7.app.AlertDialog;

```

```

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import java.util.Arrays;

public class Diagnostic extends AppCompatActivity implements View.OnClickListener {

    EditText edittext_orina, edittext_micciones, edittext_presion, edittext_diabetes,
edittext_medicina, edittext_acido_urico, edittext_reumaticas, edittext_renal, edittext_quistes,
edittext_ultrasonido;
    Button btnEnviar;
    private String[] respuestas = {"Si", "si", "SI"};
    private String[] respuestas1 = {"Aspirina", "aspirina", "ASPIRINA"};
    private String[] respuestas2 = {"Aines", "aines", "AINES"};
    private String[] respuestas3 = {"Antibioticos", "antibioticos", "ANTIBIOTICOS"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_diagnostic);

        edittext_orina = (EditText) findViewById(R.id.edittext_orina);
        edittext_micciones = (EditText) findViewById(R.id.edittext_micciones);
        edittext_presion = (EditText) findViewById(R.id.edittext_presion);
        edittext_diabetes = (EditText) findViewById(R.id.edittext_diabetes);
        edittext_medicina = (EditText) findViewById(R.id.edittext_medicina);
        edittext_acido_urico = (EditText) findViewById(R.id.edittext_acido_urico);
        edittext_reumaticas = (EditText) findViewById(R.id.edittext_reumaticas);
        edittext_renal = (EditText) findViewById(R.id.edittext_renal);
        edittext_quistes = (EditText) findViewById(R.id.edittext_quistes);
        edittext_ultrasonido = (EditText) findViewById(R.id.edittext_ultrasonido);
        btnEnviar = (Button) findViewById(R.id.btnEnviar);
        btnEnviar.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        int orina = Integer.parseInt(edittext_orina.getText().toString());
        int micciones = Integer.parseInt(edittext_micciones.getText().toString());
        int presion = Integer.parseInt(edittext_presion.getText().toString());
        int diabetes, medicina, acido_urico, reumaticas, renal, quistes, ultrasonido;

        //<-----<i>orina</i>----->
        ----->
        float a = 3 - orina;
        float b = a / 3;

        float c = orina * 1;
        float h = c / 3;

        float d = 5 - orina;
        float e = d / 2;
    }
}

```

```

float f = orina - 3;
float g = f / 2;

float OM = Math.max(Math.min(1, b), Math.min(h, e));
float OMax = Math.max(OM, Math.min(g, 1));
float orinaf = OMax * 6;

//<-----micciones----->
float i = 2 - micciones;

float j = micciones - 1;

float k = 3 - micciones;

float l = micciones - 2;

float MM = Math.max(Math.min(1, i), Math.min(j, k));
float MMax = Math.max(MM, Math.min(l, 1));
float miccionesf = MMax * 3;

//<-----presion----->
float m = 130 - presion;
float n = m / 130;

float o = presion - 120;
float p = o / 10;

float q = 140 - presion;
float r = q / 10;

float s = presion - 130;
float t = s / 10;

float PM = Math.max(Math.min(1, n), Math.min(p, r));
float PMax = Math.max(PM, Math.min(t, 1));
float presionf = PMax * 6;

//<-----diabetes----->
if (Arrays.asList(respuestas).contains(edittext_diabetes.getText().toString())) {
    diabetes = 2;
} else {
    diabetes = 1;
}

if (Arrays.asList(respuestas1).contains(edittext_medicina.getText().toString())) {
    medicina = 1;
} else if (Arrays.asList(respuestas2).contains(edittext_medicina.getText().toString())){
    medicina = 2;
} else if (Arrays.asList(respuestas3).contains(edittext_medicina.getText().toString())){
    medicina = 3;
} else {
    medicina = 1;
}

```

```

if (Arrays.asList(respuestas).contains(edittext_acido_urico.getText().toString())) {
    acido_urico = 2;
} else {
    acido_urico = 1;
}

if (Arrays.asList(respuestas).contains(edittext_reumaticas.getText().toString())) {
    reumaticas = 2;
} else {
    reumaticas = 1;
}

if (Arrays.asList(respuestas).contains(edittext_renal.getText().toString())) {
    renal = 5;
} else {
    renal = 1;
}

if (Arrays.asList(respuestas).contains(edittext_quistes.getText().toString())) {
    quistes = 2;
} else {
    quistes = 1;
}

if (Arrays.asList(respuestas).contains(edittext_ultrasonido.getText().toString())) {
    ultrasonido = 1;
} else {
    ultrasonido = 2;
}

float Salida = orinaf + miccionesf + presionf + diabetes + medicina + acido_urico + reumaticas +
renal + quistes + ultrasonido;

double u = 16.5-Salida;
double v1 = u/4.5;

double w = Salida-12;
double x = w/4.5;

double y = 33-Salida;
double z = y/16.5;

double a1 = Salida-16.5;
double b1 = a1/16.5;

double SM = Math.max(Math.min(1,v1), Math.min(x,z));
double SMax = Math.max(SM, Math.min(b1,1));
double Salidaf = SMax * 100;

AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setMessage(" ")
    .setTitle(" " + Salidaf + "%")
    .setCancelable(false)
    .setNeutralButton("Acceptar", new DialogInterface.OnClickListener(){
        public void onClick(DialogInterface dialog, int id) {

```

```

        dialog.cancel();
    }
});
AlertDialog alert = builder.create();
alert.show();
}
}

```

IntroActivity

```

package com.aletvia.appintro;

import android.support.design.widget.TabLayout;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Button;

import java.util.ArrayList;
import java.util.List;

public class IntroActivity extends AppCompatActivity {

    private ViewPager screenPager;
    IntroViewPagerAdapter introViewPagerAdapter;
    TabLayout tabIndicator;
    Button button;
    int position = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_intro);

        tabIndicator = findViewById(R.id.tab_indicator);

        List<ScreenItem> mList = new ArrayList<>();
        mList.add(new ScreenItem("CÁLCULOS RENALES", "Depósitos sólidos de minerales y sales ácidas que se agrupan en la orina concentrada. Pueden ser dolorosos cuando pasan por el tracto urinario, pero no suelen ocasionar daños permanentes.", R.drawable.icon_calculos));
        mList.add(new ScreenItem("CÁNCER DE RIÑÓN", "Comienza en el revestimiento de los pequeños conductos del riñón. Los factores de riesgo incluyen el tabaquismo y la obesidad.", R.drawable.icon_cancer));
        mList.add(new ScreenItem("INSUFICIENCIA RENAL", "Afección que provoca que los riñones pierdan la capacidad de eliminar los desechos y equilibrar los fluidos.", R.drawable.icon_insuficiencia));
        mList.add(new ScreenItem("GLOMERULONEFRITIS", "Inflamación de los pequeños filtros de los riñones (glomérulos). Estos eliminan el exceso de líquido, los electrolitos y los desechos del torrente sanguíneo, y los hacen pasar a la orina.", R.drawable.icon_glomerulonefritis));
        mList.add(new ScreenItem("INFECCIONES DEL RIÑÓN", "Sacos no cancerosos que contienen un líquido acuoso. La mayoría de las personas con este trastorno padecen de insuficiencia renal cuando tienen alrededor de 67 años.", R.drawable.icon_infeccion));
        mList.add(new ScreenItem("POLIQUISTOSIS RENAL", "Puede ser el resultado de accidentes con vehículos motorizados, riñas, caídas accidentales, causas iatrogénicas y por

```

```

deportes de contacto.", R.drawable.icon_poliquistosis));
    mList.add(new ScreenItem("LESIÓN RENAL AGUDA", "Infección de las vías urinarias que,
generalmente, comienza en la uretra o en la vejiga y sube hasta uno o ambos riñones. Si no se
trata de forma adecuada puede causar daño permanente a los riñones.", R.drawable.icon_lesion));
    mList.add(new ScreenItem("TRAUMATISMO RENAL", "Afección en la que los riñones
dejan de filtrar los residuos de la sangre repentinamente. Es común en personas que sufren de
enfermedades críticas y que ya están hospitalizadas.", R.drawable.icon_traumatismo));
    mList.add(new ScreenItem("SÍNDROME NEFRÓTICO", "Enfermedad del riñón que
ocasiona que el cuerpo elimine demasiadas proteínas en la orina. Suele aparecer como resultado
del daño en los pequeños vasos sanguíneos de los riñones que filtran los desechos y el exceso de
agua de la sangre.", R.drawable.icon_nefrotico));

    screenPager = findViewById(R.id.screen_viewpager);
    introViewPagerAdapter = new IntroViewPagerAdapter(this, mList);
    screenPager.setAdapter(introViewPagerAdapter);

    tabIndicator.setupWithViewPager(screenPager);
}
}

```

IntroViewPagerAdapter

```

package com.aletvia.appintro;

import android.content.Context;
import android.support.annotation.NonNull;
import android.support.v4.view.PagerAdapter;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import java.util.List;

public class IntroViewPagerAdapter extends PagerAdapter {

    Context mContext;
    List<ScreenItem> mListScreen;

    public IntroViewPagerAdapter(Context mContext, List<ScreenItem> mListScreen) {
        this.mContext = mContext;
        this.mListScreen = mListScreen;
    }

    @NonNull
    @Override
    public Object instantiateItem(@NonNull ViewGroup container, int position) {
        LayoutInflater inflater = (LayoutInflater)
mContext.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View layoutScreen = inflater.inflate(R.layout.layout_screen, null);

        ImageView imgSlide = layoutScreen.findViewById(R.id.intro_img);
        TextView title = layoutScreen.findViewById(R.id.intro_title);
        TextView description = layoutScreen.findViewById(R.id.intro_description);
    }
}

```

```

title.setText(mListScreen.get(position).getTitle());
description.setText(mListScreen.get(position).getDescription());
imgSlide.setImageResource(mListScreen.get(position).getScreenImg());

container.addView(layoutScreen);

return layoutScreen;
}

@Override
public int getCount() {
return mListScreen.size();
}

@Override
public boolean isViewFromObject(@NonNull View view, @NonNull Object o) {
return view == o;
}

@Override
public void destroyItem(@NonNull ViewGroup container, int position, @NonNull Object object) {
container.removeView((View)object);
}
}

```

ScreenItem
<pre> package com.aletvia.appintro; public class ScreenItem { String Title, Description; int ScreenImg; public ScreenItem(String title, String description, int screenImg) { Title = title; Description = description; ScreenImg = screenImg; } public void setTitle(String title) { Title = title; } public void setDescription(String description) { Description = description; } public void setScreenImg(int screenImg) { ScreenImg = screenImg; } public String getTitle() { return Title; } </pre>

```

}

public String getDescription() {
    return Description;
}

public int getScreenImg() {
    return ScreenImg;
}
}

```

FinalActivity
<pre> package com.aletvia.appintro; import android.app.DatePickerDialog; import android.content.Intent; import android.support.v4.app.DialogFragment; import android.support.v7.app.AppCompatActivity; import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.DatePicker; import android.widget.ImageView; import android.widget.TextView; import java.text.DateFormat; import java.util.Calendar; public class FinalActivity extends AppCompatActivity implements DatePickerDialog.OnDateSetListener{ ImageView ivSalir; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_final); Button button = (Button) findViewById(R.id.button); button.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View v) { DialogFragment datePicker = new DatePickerFragment(); datePicker.show(getSupportFragmentManager(), "Seleccionar fecha"); } }); ivSalir = (ImageView) findViewById(R.id.ivSalir); ivSalir.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View v) { Intent intent = new Intent(Intent.ACTION_MAIN); intent.addCategory(Intent.CATEGORY_HOME); intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK); startActivity(intent); } }); } } </pre>

```

    });
}

@Override
public void onDateSet(DatePicker view, int year, int month, int dayOfMonth){
    Calendar c = Calendar.getInstance();
    c.set(Calendar.YEAR, year);
    c.set(Calendar.MONTH, month);
    c.set(Calendar.DAY_OF_MONTH, dayOfMonth);
    String currentDateString = DateFormat.getDateInstance(DateFormat.FULL).format(c.getTime());

    TextView textView = (TextView) findViewById(R.id.textView);
    textView.setText(currentDateString);
}
}

```

DatePickerFragment

```

package com.aletvia.appintro;

import android.app.DatePickerDialog;
import android.app.Dialog;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.app.DialogFragment;

import java.util.Calendar;

public class DatePickerFragment extends DialogFragment {

    @NonNull
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState){
        Calendar c = Calendar.getInstance();
        int year = c.get(Calendar.YEAR);
        int month = c.get(Calendar.MONTH);
        int day = c.get(Calendar.DAY_OF_MONTH);
        return new DatePickerDialog(getActivity(), (DatePickerDialog.OnDateSetListener) getActivity(),
        year, month, day);
    }
}

```