

EVALUACIÓN Y MANTENIMIENTO DEL AMBIENTE INFORMÁTICO PARA LA  
INTERACCIÓN MÉDICO-PACIENTE EN LA INTERVENCIÓN DE FACTORES DE  
RIESGO DE HIPERTENSIÓN ARTERIAL “RIESGO HTA 2.0”

YADDY KATHERIN ROJAS SANTOS  
JENNY KATHERINE SARMIENTO ORTIZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA

2020

EVALUACIÓN Y MANTENIMIENTO DEL AMBIENTE INFORMÁTICO PARA LA  
INTERACCIÓN MÉDICO-PACIENTE EN LA INTERVENCIÓN DE FACTORES DE  
RIESGO DE HIPERTENSIÓN ARTERIAL “RIESGO HTA 2.0”

YADDY KATHERIN ROJAS SANTOS  
JENNY KATHERINE SARMIENTO ORTIZ

Trabajo de grado para optar al título de  
INGENIERO DE SISTEMAS

Director  
HUGO HERNANDO ANDRADE SOSA  
Magíster en informática

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA

2020

## **DEDICATORIA**

### **Yaddy Katherin Rojas Santos**

Primero que todo quiero agradecerle a Dios por todo lo que tiene planeado para cada uno de nosotros y brindarnos las oportunidades en nuestros proyectos de vida.

A toda mi familia, mamá, papá, hermano, tías, tíos, primos, por el apoyo y amor incondicional durante mi estudio, logros durante mi carrera y mi futuro.

A mi compañera de lucha katherine, nosotras sabemos por lo que tuvimos que pasar durante la elaboración de este proyecto y nuestra carrera para cumplir una de tantas metas presentes en nuestra vida, GRACIAS

A mis amigos que siempre estuvieron ahí dando me un apoyo enorme durante la carrera, por sus críticas para ser una persona mejor en la vida y en el futuro que me queda por delante.

Al grupo SIMON por aceptarnos y formarnos como investigadoras y profesionales en la elaboración de este proyecto de grado.

## DEDICATORIA

**Jenny Katherine Sarmiento Ortiz**

**A dios y a la vida** Porque me trajeron hasta aquí y todo el proceso estuvo lleno de enseñanzas y cosas positivas para mi vida, personal y laboral.

**A mi madre y padre** Porque gracias a ellos tuve la oportunidad de vivir esta experiencia y atravesar todo este proceso, merecen esto más que yo.

**A mi compañera** porque estuvo conmigo a pesar de todas las circunstancias y gracias a ella pude llegar hasta aquí.

**A mi mejor amiga** porque me apoyo desde el inicio y estuvo conmigo en todos los momentos buenos y malos durante este proceso. Gracias por estar conmigo siempre y no dejarme caer.

## **Agradecimientos**

A la Universidad Industrial de Santander y a la escuela de ingeniería de sistemas por todos los espacios y apoyos que nos brindaron para el crecimiento personal y profesional de sus estudiantes.

Al profesor Hugo Hernando Andrade Sosa, por permitirnos hacer parte de su grupo de investigaciones SIMON, por su tiempo y orientación para la finalización de este proyecto de grado.

Al ingeniero y amigo Juan Daniel Hernández, por su disposición, tiempo y conocimiento que fueron de gran ayuda para el desarrollo de este proyecto.

A nuestro compañero Brandon Vásquez, porque en la última etapa de este proyecto, nos ayudó sin ningún reparo.

Al ingeniero Eduardo Guerra asistente del grupo de Simón que nos colaboró en la instalación de nuestro proyecto al servidor del grupo.

A todas las personas maravillosas que conocimos en esta etapa de la vida, nuestros amigos y compañeros que hicieron parte de este logro, porque incluso una palabra hizo que esto se pudiera llevar a cabo.

## CONTENIDO

	pág.
INTRODUCCIÓN .....	14
1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA.....	16
2. ALCANCE .....	19
3. OBJETIVOS.....	20
3.1 OBJETIVOS GENERALES .....	20
3.2 OBJETIVOS ESPECÍFICOS.....	20
4. MARCO DE REFERENCIA.....	21
4.1 TIC EN EL SECTOR DE LA SALUD.....	21
4.2 PENSAMIENTO SISTÉMICO Y DINÁMICA DE SISTEMAS .....	22
4.3 INGENIERÍA DEL SOFTWARE .....	22
4.4 HIPERTENSIÓN ARTERIAL.....	23
4.5 FACTORES DE RIESGO HTA .....	23
4.6 PRUEBAS DE SOFTWARE.....	24
4.7 MANTENIMIENTO DE SOFTWARE.....	25
4.8 RIESGO HTA VERSION 1.0.....	25
4.9 ESTUDIO DE FRAMINGHAM.....	26
5. METODOLOGÍA .....	27
6. EVALUACIÓN PRIMERA VERSIÓN.....	29
6.1 REVISIÓN DE MODELO EVOLUCIÓN .....	29
6.2 PRUEBAS TÉCNICAS VERSIÓN 1.0.....	31

6.3 PRUEBAS DE RECEPTIVIDAD .....	32
6.4 CONCLUSIONES DE LA EVALUACIÓN .....	34
7. DISEÑO .....	36
7.1 DESCRIPCIÓN GENERAL .....	36
7.2 MODELO EVOLUCIÓN .....	38
7.3 FUNCIONALIDADES .....	38
7.3.1 Especificación de requerimientos .....	39
7.4 DIAGRAMAS UML .....	43
7.4.1 Diagrama entidad relación .....	44
7.4.2 Diagrama de casos de uso .....	45
7.4.3 Diagrama de actividades.....	47
7.5 DESCRIPCIÓN CASOS DE USO .....	49
8. DESARROLLO .....	60
8.1 MODELO VISTA CONTROLADOR .....	60
8.2 BACKEND.....	62
8.2.1 Spring Framework.....	62
8.2.2 Spring Boot .....	63
8.2.3 Spring Tool Suite .....	64
8.2.4 Estructura del back .....	64
8.3 FRONTEND .....	67
8.3.1 Angular.....	67
8.3.2 Node Js.....	69
8.3.3 Características y componentes de angular .....	69
8.3.4 Estructura del frontend.....	72
8.3.5 Resultado Final .....	73
9. PRUEBAS.....	74

10. INTEGRACIÓN .....	75
11. CONCLUSIONES .....	78
12. TRABAJO FUTURO.....	79
BIBLIOGRAFÍA.....	81

## LISTA DE TABLAS

	pág.
Tabla 1. Plantilla para la definición de requerimientos funcionales.....	39
Tabla 2. Plantilla de requerimientos No funcionales .....	39
Tabla 3. RF01-Registrar paciente .....	39
Tabla 4. RF02-Iniciar sesión por roles .....	40
Tabla 5. RF03-Hacer simulaciones.....	40
Tabla 6. RF04-Registrar visitas.....	41
Tabla 7. RF05-Editar paciente .....	41
Tabla 8. RF06-Listar pacientes.....	41
Tabla 9. RF07-Historial de visitas .....	42
Tabla 10. RNF01 - Interfaz de usuario.....	42
Tabla 11. RNF02 - Seguridad .....	42
Tabla 12. RNF03 - Disponibilidad .....	43
Tabla 13. RNF05 - Desempeño .....	43
Tabla 14. Plantilla casos de uso .....	49
Tabla 15. CU1 Iniciar sesión .....	49
Tabla 16. CU2 Registrarse.....	51
Tabla 17. CU3 Editar paciente.....	52
Tabla 18. CU4 Agregar visita .....	54
Tabla 19. CU5 Historial de visitas .....	56
Tabla 20. CU6 Simular.....	57
Tabla 21. CU7 Cerrar sesión .....	58

## LISTA DE FIGURAS

	<b>Pág.</b>
Figura 1. Metodología en cascada .....	28
Figura 2. Mapa de sectores del modelo final .....	29
Figura 3. Modelo para el riesgo de la hipertensión arterial en evolución .....	30
Figura 4. Pruebas de instalación en los distintos SO .....	32
Figura 5. Diagrama entidad relación .....	45
Figura 6. Diagrama de caso de usos .....	46
Figura 7. Diagrama de actividades del paciente .....	47
Figura 8. Diagrama de actividades del visitador .....	48
Figura 9. Modelo vista controlador .....	61
Figura 10. Arquitectura Spring .....	63
Figura 11. Estructura del backend .....	65
Figura 12. Logo angular .....	68
Figura 13. Angular cli .....	69
Figura 14. Estructura del front.....	72

## **LISTA DE ANEXOS**

**(Ver anexos adjuntos y pueden ser visualizados en la base de datos de la biblioteca UIS)**

- Anexo A. Informe pruebas del modelo HTA 1.0
- Anexo B. Informe pruebas software HTA 1.0
- Anexo C. Análisis de software HTA 1.0
- Anexo D. Pruebas de usuario HTA 1.0
- Anexo E. Presentación “RIESGO HTA2” - UISALUD
- Anexo F. Pruebas de software HTA 2.0
- Anexo G. Pruebas de usuario HTA 2.0
- Anexo H. Informe de reuniones UISALUD

## RESUMEN

**TITULO:** EVALUACIÓN Y MANTENIMIENTO DEL AMBIENTE INFORMÁTICO PARA LA INTERACCIÓN MÉDICO-PACIENTE EN LA INTERVENCIÓN DE FACTORES DE RIESGO DE HIPERTENSIÓN ARTERIAL<sup>1</sup> “RIESGO HTA 2.0”

**AUTORES:** YADDY KATHERIN ROJAS SANTOS  
JENNY KATHERINE SARMIENTO ORTIZ<sup>2</sup>

**PALABRAS CLAVE:** Evaluación de software, Hipertensión arterial, Informática de la salud, Dinámica de sistemas

### DESCRIPCIÓN:

Para la detección, evaluación y tratamiento de la hipertensión, el seguimiento constante y estricto a los pacientes es clave, teniendo en cuenta que esto determina un aumento en el riesgo de padecer complicaciones cardiovasculares. En el grupo de investigaciones SIMON se plantea una propuesta que vincula al paciente con su estado de salud actual y además le brinde un soporte a la entidad encargada del paciente, en este caso es UISALUD. Teniendo base la primera versión desarrollada en modalidad trabajo de grado titulado *Ambiente Informático Para la Interacción Médico-Paciente en la Intervención de Factores de Riesgo de Hipertensión Arterial*, en el que por medio de la evaluación además de estrechar la relación médico paciente, permite también que el paciente por sus propios medios tenga consciencia de su estado de salud y tenga la posibilidad de simular su tendencia al riesgo por sí mismo. Además, por medio de una aplicación web será posible hacer el registro de la información percibida en visitas realizadas por un profesional de salud a los respectivos hogares, con un pronóstico cualitativo del riesgo donde se plantee una discusión dentro del entorno familiar, pues es de gran importancia identificar este tipo de enfermedades las cuales están asociadas de alguna manera al desarrollo económico y progreso social.

---

<sup>1</sup> Trabajo de grado

<sup>2</sup> Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de sistemas e informática. Director Hugo Hernando Andrade Sosa. Magíster en informática.

## ABSTRACT

**TITLE:** EVALUATION AND MAINTENANCE OF THE COMPUTER ENVIRONMENT FOR MEDICAL-PATIENT INTERACTION IN THE INTERVENTION OF RISK FACTORS OF ARTERIAL HYPERTENSION<sup>3</sup> “RIESGO HTA 2.0”

**AUTHOR:** YADDY KATHERIN ROJAS SANTOS  
JENNY KATHERINE SARMIENTO ORTIZ<sup>4</sup>

**KEYWORDS:** Arterial hypertension, Software evaluation, Software maintenance, Information technology, Dynamic of systems.

### DESCRIPTION:

For the detection, evaluation and treatment of hypertension, constant and strict monitoring of patients is key, bearing in mind that this determines an increased risk of cardiovascular complications. In the SIMON research group, a proposal is made that links the patient to their current health status and also provides support to the entity in charge of the patient, in this case, UISALUD. Based on the first version developed in a degree work modality entitled Computer Environment for the Doctor-Patient Interaction in the Intervention of Risk Factors of Arterial Hypertension, in which, through evaluation, in addition to narrowing the doctor-patient relationship, it also allows that the patient by his own means is aware of his state of health and has the possibility to simulate his tendency to risk by himself. In addition, through a web application it will be possible to record the information perceived in visits made by a health professional to the control homes, with a qualitative forecast of risk where a discussion arises within the family environment, since it is of great importance to identify this type of diseases which are associated in some way with economic development and social progress.

---

<sup>3</sup> Bachelor thesis

<sup>4</sup> Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de sistemas e informática. Director Hugo Hernando Andrade Sosa. Magíster en informática.

## INTRODUCCIÓN

Un análisis del Observatorio de Salud Pública de Santander muestra que la principal causa de muerte en el departamento son las enfermedades cardiovasculares<sup>5</sup>, y la hipertensión arterial (HTA) es una de las principales causas de enfermedades cardiovasculares (ECV<sup>6</sup>).

Con el pasar de los años, la HTA no solo afecta órganos como el corazón causando infartos o falla cardiaca, también afecta el cerebro causándole enfermedad cerebrovascular, los riñones llevándolos a entrar en falla renal, o la vista causando inclusive ceguera. Dentro de las principales causas de ERC (enfermedad renal crónica) en el país se encuentra la hipertensión y otros tipos de enfermedades cardiovasculares. Por lo tanto, es importante crear conciencia en la población, que conozcan la importancia de prevenir y manejar estas enfermedades por medio del control de los factores de riesgo que son modificables<sup>7</sup>. Por esta razón se busca fomentar un mejor estilo de alimentación y vida para los pacientes que aún no sufren dicha enfermedad y para los que tienen herencia a esta.

Con la intención de aportar a los programas de UISALUD denominados “Programa de Todo Corazón (PTC)” y “Programa de Riesgo Cardiovascular (RCV)” enfocados

---

<sup>5</sup> OBSERVATORIO DIGITAL MUNICIPAL [sitio web], INFORMACION PUBLICA, Bucaramanga; ALCALDÍA DE BUCARAMANGA. Información pública, salud y ambiente, causas de mortalidad Disponible en: <http://observatorio.bucaramanga.gov.co/index.php/informacion-publica/> pág. 2

<sup>6</sup> PORTAL WEB MINISTERIO DE SALUD Y PROTECCION SOCIAL [sitio web], Biblioteca digital. Disponible en: <https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/VS/PP/ENT/dia-mundial-hipertension-2017.pdf> pág. 3

<sup>7</sup> PORTAL WEB MINISTERIO DE SALUD Y PROTECCION SOCIAL [sitio web], Biblioteca digital. Disponible en: <https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/VS/ED/PSP/asis-colombia-2016.pdf> pág. 3

en la prevención, diagnóstico temprano, manejo adecuado y disminución de complicaciones futuras en enfermedades cardiovasculares, en el grupo SIMON de investigaciones proponen un enfoque dinámico sistémico para identificar las relaciones estructurales en el fenómeno HTA<sup>8</sup> y modelar una dinámica de búsqueda de acuerdos en la relación médico-paciente, apoyada en un sistema de pronóstico del impacto que pueda lograrse para dar cambios al modo de vida del paciente.

Con el propósito planteado, se desarrolló la primera versión del software riesgo HTA que tuvo como resultado un software de escritorio para la interacción médico-paciente, sin embargo, en este proyecto se propuso evaluar ese software, entendiéndose por evaluación, realizar pruebas de usuario y probar que ese software funciona correctamente, y teniendo en cuenta esa evaluación y las recomendaciones realizadas en la primera versión, tomar decisiones con respecto a ese modulo medico paciente, además desarrollar dos módulos web, uno para el visitador y uno para el paciente, todo esto será descrito a lo largo de este documento, siempre teniendo en cuenta que se desarrolló con la intención de servir como prototipo base de un proyecto de investigación que está surgiendo desde el grupo de investigación SIMON a UISALUD.

---

<sup>8</sup> ANDRADE SOSA, Hugo Hernando. *et al.* Pensamiento sistémico diversidad en búsqueda de unidad. Bucaramanga. Ediciones Universidad Industrial de Santander, 2007.

## 1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

Las enfermedades relacionadas con el estilo de vida comparten factores de riesgo que hacen referencia a tres (3) hábitos que han sido prolongados a lo largo de la vida: tabaquismo, dietas poco saludables e inactividad física<sup>9</sup>, y desarrollan enfermedades crónicas comúnmente relacionadas con el corazón.

En un país como Colombia donde el servicio de salud, centrado en lo asistencial y los medicamentos es crítico, los pacientes no cuentan con un acompañamiento adecuado para el control de enfermedades en particular la hipertensión arterial que en gran parte dependen de los buenos hábitos que tenga el paciente, es por esto que frente a posibles recomendaciones del médico, los pacientes tienen dos alternativas: llegar a un acuerdo con el médico por medio de directrices que contribuyan al mejoramiento del bienestar físico o por otro lado hacer caso omiso a las recomendaciones y correr un alto riesgo de que con el tiempo no solo se manifieste una hipertensión arterial creciente si no alguna enfermedad coronaria mucho más grave, además en el caso de un común acuerdo médico-paciente hay que tener en cuenta que no siempre se cumple, debido a que el paciente desconoce en todo momento su estado de salud, el cumplimiento de los buenos hábitos se hace más tedioso para el paciente, al ser una enfermedad que no se muestra de forma inmediata ante el incumplimiento de una “directriz” recomendada por el médico, el paciente pierde interés en su salud y en la intervención temprana de HTA, todo esto hace que el vincular al paciente con este continuo proceso sea una tarea de motivación y apoyo por parte del médico y de la entidad de salud encargada, se trata de establecer una relación estrecha entre estos, de forma que el paciente sea

---

<sup>9</sup> CARRILLO, Leidy M. & PORTILLA, Yeny P. Ambiente informático para la interacción médico-paciente en la intervención de factores de riesgo de hipertensión arterial Versión 1.0. Trabajo de grado, Universidad Industrial de Santander, Bucaramanga. 2017. Pág. 28

educado y por otra parte comprenda que ciertos factores intervienen de forma negativa en su salud, aunque él no lo note, y el médico de alguna forma pueda recordar e intervenir en este cambio mostrándole al paciente las posibles consecuencias y beneficios que trae comprometerse con este proceso.

Entorno a esta problemática en el grupo SIMON de investigaciones en Modelamiento y Simulación, perteneciente a la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander – UIS, fue desarrollada la primera versión de una herramienta informática orientada a apoyar la atención al problema del riesgo de sufrir hipertensión y en la cual se fomenta la prevención de factores de riesgo de la misma por medio de una relación médico-paciente de forma que ambos puedan tener un mejor control acerca de cómo están actuando sobre el sistema esas directrices que el médico considera que son de vital importancia en el proceso de prevención y mejora.

La versión 1.0 del software fue un primer prototipo que no fue probado con suficiente variedad de usuarios, solo se realizaron algunas pruebas de operación con pocos usuarios, es por eso que es importante evaluarlo y con distintas condiciones que permitan identificar y corregir errores, por lo que un software que está encaminado a aportar en el sector de la salud demanda una evaluación mucho más estricta y pruebas de usuario con variedad de escenarios para que se puedan considerar todas las posibilidades, es por esto que es importante atender esa necesidad para tener un software más eficiente y con mayor acabado contemplando distintas opiniones.

Si tenemos en cuenta la problemática mencionada anteriormente en una comunidad específica, como la comunidad de la Universidad Industrial de Santander la cual es atendida por la entidad prestadora UISALUD, es importante tener en cuenta que el médico y otros profesionales de la salud requieren de una herramienta que de una u otra forma facilite el proceso de aprendizaje y permita que la interacción médico

paciente sea más estrecha, incluso se propuso una posibilidad en la que se realicen visitas directamente al hogar del paciente para hacer un seguimiento estricto a la salud y condiciones de este, de esta manera no solo el médico por medio de una cita médica podrá discutir con el paciente acerca de su salud y como los buenos hábitos de vida son parte fundamental del proceso, si no que el profesional desde el mismo entorno del paciente cree esta discusión familiar e incluso recolecte información que pueda ser importante y pueda ayudar a controlar o evaluar otras enfermedades, es por esto que en este proyecto se planeó crear dos módulos nuevos tanto para el paciente como para el visitador, que sirvan como prototipo para que la entidad de salud evalúe y se pueda desarrollar el proyecto de investigación que está en marcha partiendo de una posible opción, sería un primer paso en el marco de un programa de salud que contemple involucrar en mayor medida al paciente y su entorno familiar.

## 2. ALCANCE

Este proyecto propone un prototipo inicial de una interfaz web que sirva como base para crear una herramienta que pueda ser utilizada por una entidad de salud, para hacer el seguimiento oportuno a la salud del paciente e impactar principalmente la atención primaria que se da en estas entidades.

Teniendo en cuenta la evaluación de la primera versión de este proyecto se decidió sobre que plataformas trabajar y como formular el diseño de estos nuevos módulos (paciente y visitador), que básicamente cumplen la misma función que el software de escritorio que fue desarrollado como primera versión, en este sentido más allá de enfocarnos en realizar un nuevo prototipo del modelo que nos muestra la tendencia del riesgo a sufrir de hipertensión, que es base fundamental del proyecto pero que requiere la ayuda de profesionales en el área de salud para ser desarrollado, se busca mostrar por medio de una interfaz web que integra el modelado sistémico, que este tipo de herramientas pueden ser de mucha utilidad dentro del área de salud y que dependiendo de las herramientas que se usan para el desarrollo de esta, una posible integración con el sistema de salud es factible, sin embargo, no se busca realizar esa integración de manera rigurosa si no procurar que se pueda realizar, ya que este proyecto está encaminado a usarse como modelo web y no como versión final del proyecto de investigación que se está poniendo en marcha desde el grupo de investigación SIMON y que en una futura versión se podría enfocar más en el diseño del modelo, ajustándolo a las necesidades de la entidad de salud y así proponer un diseño más general basado en el que se plantea aquí, con nuevas funcionalidades e incluso desarrollado bajo la metodología de diseño por prototipos.

### **3. OBJETIVOS**

#### **3.1 OBJETIVOS GENERALES**

Evaluar la primera versión del software “RiesgoHTA 1.0” desarrollado para facilitar la relación médico-paciente en la intervención de factores de riesgo de hipertensión arterial. Mediante la apreciación y pruebas orientadas por expertos, junto con la evaluación desde la ingeniería de software; definiendo los requerimientos de una nueva versión y desarrollando un prototipo de esta.

#### **3.2 OBJETIVOS ESPECÍFICOS**

- Evaluar el software “RiesgoHTA” para la interacción médico-paciente en la intervención de factores de riesgo de hipertensión arterial mediante la apreciación y experiencias orientadas por expertos de UISALUD, junto con pruebas técnicas de ingeniería de software.
- Hacer un mantenimiento al software “RiesgoHTA”, según los resultados de la evaluación.
- Desarrollar un módulo para que el usuario (paciente) pueda considerar los efectos de alternativas frente a su problema de salud.
- Desarrollar un módulo del visitador familiar, quien es la persona encargada de realizar la visita en casa al paciente.
- Integrar con el sistema de atención general de la salud del paciente.

## 4. MARCO DE REFERENCIA

### 4.1 TIC EN EL SECTOR DE LA SALUD

El sector de la salud es un tema fundamental en cualquier sociedad y Colombia no es la excepción, teniendo en cuenta que la cobertura de este sector ha ido aumentando con los años, el impacto que pueden tener las TIC sobre esto puede tener cualquier magnitud.

Software Médico, sistemas de gestión institucional, acceso a servicios de información para profesionales y paciente, y soporte de comunicación a las actividades asistenciales, médicas y quirúrgicas, son 3 de los escenarios donde se puede observar la implementación de TICs en este sector<sup>10</sup>.

Actualmente es importante que la tecnología y la sociedad avancen paralelamente; con la tecnología podemos mejorar este campo en varios aspectos: reducir costos, mejorar la calidad de la atención, innovar entre otros, por ejemplo el nodo de salud en Colombia fomenta la creación de productos y servicios para dar soluciones al sector de la salud, a partir de las TIC, este es un proceso que se encuentra en auge, sin embargo, lo que sí está claro es que la tecnología representa un gran avance no solo en salud si no en cualquier campo debido a la transversalidad que representa la tecnología y a todos los beneficios que se pueden obtener de esta.

---

<sup>10</sup> AVELLA MARTINEZ, Laura Yaneth & PARRA RUIZ, Paola P. Tecnologías de la información y la comunicación (TICS) en el sector salud. Trabajo de investigación, Universidad Nacional de Colombia, Bogotá. 2013

## 4.2 PENSAMIENTO SISTÉMICO Y DINÁMICA DE SISTEMAS

El pensamiento sistémico intenta resolver problemas dentro de un sistema complejo, es una disciplina de enfoque interdisciplinario o transversal a las demás áreas, ya que depende de su aplicación y puede estar relacionada con la economía, industria, salud, entre otros. Está basada en el trabajo de la teoría general de sistemas de Ludwing Von Bertalanffy<sup>11</sup>. Su principal característica es que no intenta aislar las variables de un escenario y estudiarlo por aparte si no estudiar todo como un todo, contrario a lo que se hace en el método científico tradicional, además estudia posibles soluciones y sus efectos a corto y largo plazo.

## 4.3 INGENIERÍA DEL SOFTWARE

Es una disciplina que integra matemáticas y ciencias de la computación, así mismo con un enfoque sistémico se orienta para el desarrollo, operación y mantenimiento de software, sus actividades principales son especificación, desarrollo, evolución y mantenimiento de software. También es una disciplina de enfoque interdisciplinario o transversal a las demás áreas, ya que depende de su aplicación, puede estar relacionada con la economía, industria, salud, entre otros <sup>12</sup>. Trasciende la actividad de programar, pues incluye el análisis previo del escenario, el diseño, el desarrollo y las pruebas, además el desarrollo implica lo que conocemos como ciclo de vida del software.

---

<sup>11</sup> Unidad académica Yucatán, Laboratorio de biología, Teoría general de sistemas, [En línea], Disponible en: [http://www.sisal.unam.mx/labeco/LAB\\_ECOLOGIA/Ecologia\\_y\\_evolucion\\_files/XI.%20TEORIA%20GENERAL%20DE%20SISTEMAS.pdf](http://www.sisal.unam.mx/labeco/LAB_ECOLOGIA/Ecologia_y_evolucion_files/XI.%20TEORIA%20GENERAL%20DE%20SISTEMAS.pdf)

<sup>12</sup> PRESSMAN. Roger. Ingeniería del software. Un enfoque práctico. México. McGraw-Hill. 2010. Cap. 1.

#### **4.4 HIPERTENSIÓN ARTERIAL**

Es el término que se usa para referirse a la presión arterial alta, la presión arterial es una medida de la fuerza que ejercen las arterias a medida que el corazón bombea sangre. La lectura de esta se da con dos números, el número superior denota la presión sistólica (cuando el corazón late) y el número inferior la presión diastólica (cuando el corazón se relaja), y se mide en milímetros de mercurio. La hipertensión arterial no presenta síntomas claros y no se manifiesta durante mucho tiempo, sin embargo, si no se siguen las recomendaciones pertinentes puede desencadenar otros problemas más graves que pueden incluso causar la muerte<sup>13</sup>.

#### **4.5 FACTORES DE RIESGO HTA**

Ciertos hábitos, características y enfermedades aumentan la probabilidad de sufrir Hipertensión Arterial (HTA), por esto son denominados factores de riesgo HTA, dentro de estos factores se encuentran: la edad, el peso, algunos medicamentos, los hábitos poco saludables entre otros. Cabe resaltar que existen algunos factores que son modificables como los hábitos poco saludables, y hay otros que no son modificables y se deben a factores sociales o biológicos, es decir son innatos en nosotros <sup>14</sup>.

---

<sup>13</sup> Antonio, L. (2018). Programa para la detección, evaluación y tratamiento del paciente hipertenso de la región andina. [En línea] Disponible en: <http://svmi.web.ve/wh/documentos/HTA-LASH.pdf>, Prólogo

<sup>14</sup> Antonio, L. (2018). Programa para la detección, evaluación y tratamiento del paciente hipertenso de la región andina. [En línea] Disponible en: <http://svmi.web.ve/wh/documentos/HTA-LASH.pdf> Pág. 61

## 4.6 PRUEBAS DE SOFTWARE

Las pruebas software son un conjunto de actividades orientadas a comprobar determinados aspectos de un software y proveer información objetiva sobre el mismo. Existen varias clases de pruebas tales como: las estáticas son las que no corre el código de la aplicación y las dinámicas son las que si hay que correr el código de la aplicación, por otra parte, dependiendo del enfoque de las pruebas técnicas están<sup>15</sup>:

- Pruebas de caja negra: Es el componente por el cual se estudia el elemento desde las entradas y salidas que contienen, sin tener en cuenta su funcionamiento interno, es decir, se verifica que las funciones son operativas.
- Pruebas de caja blanca: Estas pruebas se centran directamente en el código, en detalles procedimentales del sistema software, es decir se busca probar que cualquier flujo de ejecución devuelve un valor correcto.
- Testing aleatorio: Son pruebas aleatorias enfocadas en distintos objetivos y etapas del software, tiene que ver con el usuario en el sentido que buscan simular el entorno e identificar posibles “errores”.

---

<sup>15</sup> Excelencia, P. (2018). Software QA - ¿Cuáles son los tipos de pruebas software?. [en Línea] Panel Sistemas. Disponible en: <https://www.panel.es/blog/software-qa-cuales-son-los-tipos-de-pruebas-software/>

## 4.7 MANTENIMIENTO DE SOFTWARE

Es una de las actividades que hacen parte de la ingeniería del software y es el proceso de mejorar y optimizar un software<sup>16</sup>.

Existen 4 tipos de servicios de mantenimiento:

1. Perfectivo: Se conoce también como mantenimiento evolutivo y mejora el software en cuanto rendimiento, usabilidad e incluso nuevos requerimientos.
2. Adaptativo: Lo que hace es adaptar el software existente a nuevas tecnologías, como nuevo sistema operativo, o nuevo entorno de base de datos entre otros.
3. Correctivo: Corrige errores que se presentaron durante la ejecución.
4. Preventivo: Trata de prevenir errores futuros o facilitar un mantenimiento futuro.

Por medio de metodologías de mantenimiento de software se puede determinar y adaptar las características de uso y actividades del software, con una serie de procesos de mantenimiento para gestionar las herramientas.

## 4.8 RIESGO HTA VERSION 1.0<sup>17</sup>

Título del proyecto: Ambiente informático para la interacción médico-paciente en la intervención de factores de riesgo de hipertensión arterial

Autores: Leidy Milena Carrillo Hernández, Yeny Patricia Portilla Hernández

Productos software.

---

<sup>16</sup> Sincows.com. (2018). Mantenimiento de Software. [En línea] Disponible en: [http://www.sincows.com/sincows/index.php?option=com\\_content&view=article&id=70&Itemid=68](http://www.sincows.com/sincows/index.php?option=com_content&view=article&id=70&Itemid=68)

<sup>17</sup> CARRILLO, Leidy M. & PORTILLA, Yeny P. Ambiente informático para la interacción médico-paciente en la intervención de factores de riesgo de hipertensión arterial Versión 1.0. Trabajo de grado, Universidad Industrial de Santander, Bucaramanga. 2017

Aplicación de escritorio: Software basado en los propuestos desde el grupo de investigación. Está enfocado solo al médico, permite registrar y hacer control a los pacientes por medio de un escenario base y un escenario propuesto, de acuerdo con los factores como la edad, dieta, frecuencia de ejercicio entre otros, este modela el estado actual de riesgo a padecer HTA o como estará su estado de riesgo cuando cambie algunos aspectos de su vida, además se encuentra un manual de instalación y un manual de usuario.

#### **4.9 ESTUDIO DE FRAMINGHAM**

El estudio de Framingham, es un estudio sobre el riesgo cardiovascular que se inició en 1948 y que aún se encuentra en marcha, este estudio es uno de los más importantes y continúa siendo referente en la actualidad, donde participa la tercera generación desde que se inició el estudio, el objetivo principal es la detección temprana de enfermedades cardíacas y el diagnóstico de manifestaciones o riesgos en personas aparentemente sanas. Además, se encuentran algunas variantes del estudio en el que se presentan los indicadores para estimar el riesgo no solo de la hipertensión arterial si no de la diabetes y la enfermedad del hígado graso, en la página oficial se presenta una calculadora de puntuación de riesgo para cada una de acuerdo al avance del estudio <sup>18</sup>.

---

<sup>18</sup> Framingham Heart Study, A project of the National Hear, Lung, and Blood Institute and Boston University, (2016), Hypertension [online]. Disponible en: <<https://framinghamheartstudy.org/fhs-risk-functions/hypertension/#>>

## 5. METODOLOGÍA

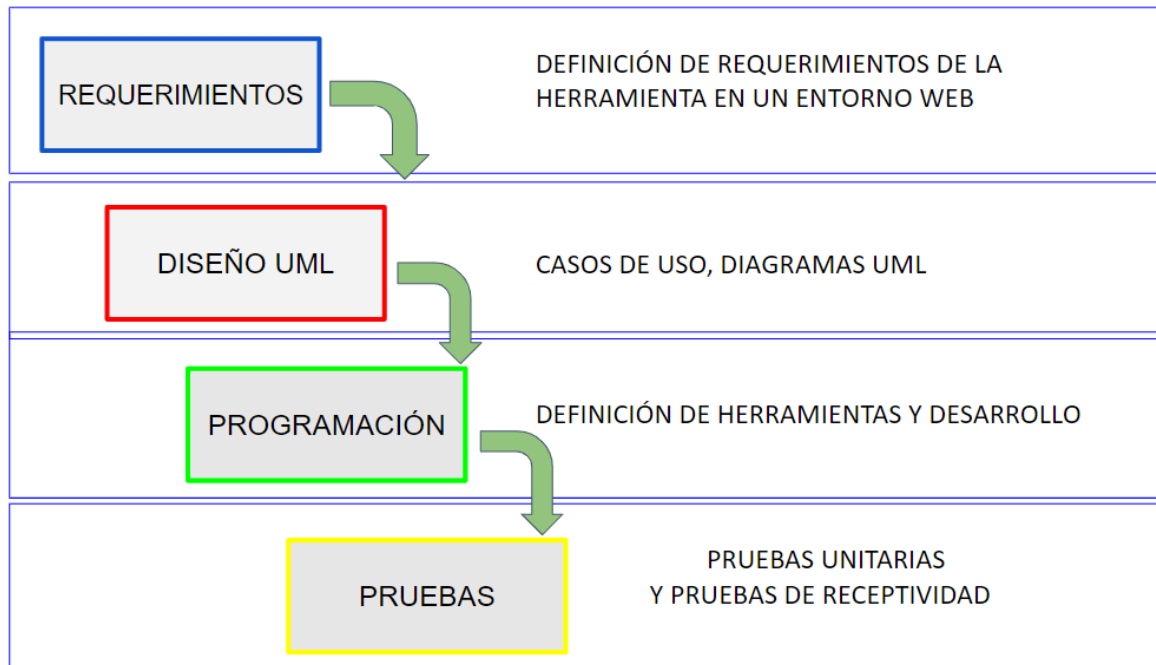
Este proyecto de grado buscaba evaluar la primera versión de “RiesgoHTA” para fortalecer la relación médico-paciente en la prevención de la hipertensión arterial, de igual forma se realizó un módulo de visitador familiar y uno del paciente con el fin de promover el autocuidado y responsabilidad personal, con lo que se compone una segunda versión, sin embargo, cabe resaltar que este proyecto es un prototipo.

Se llevó a cabo una serie de pasos que ayudaron a obtener una mejor evaluación para dar un mantenimiento adecuado a dicho software:

1. Como primera medida para obtener una evaluación se aplicaron pruebas técnicas de software que permitieron determinar falencias en la versión anterior, y permitieron tener una mejor visualización de lo que requiere el software y hacia donde debe estar enfocado el prototipo.
2. Luego de realizar dichas pruebas de evaluación y validación esto conllevó a definir los requerimientos a partir de los resultados obtenidos para tener en cuenta los aspectos finales que deben contener los módulos del usuario (paciente) y del visitador, en base a esto se definieron las herramientas que fueron usadas para el desarrollo de los módulos.
3. Se desarrolló el modelo nuevo de ambos módulos para dar un visto bueno y conceptualizar el modelo con los interesados.

Para el desarrollo de los nuevos módulos se tuvo en cuenta tal como se había propuesto la metodología de desarrollo en cascada que es un modelo lineal que emplea un diseño secuencial, que va etapa tras etapa, iniciando por la etapa de requerimientos, seguido por el diseño, el desarrollo y en este caso se ajustó para ser completado hasta la parte de pruebas o verificación:

Figura 1. Metodología en cascada



Sin embargo, debido a que este proyecto se considera un prototipo de lo que podría ser un software robusto orientado al seguimiento de la salud del paciente, se llevó a cabo solo hasta la pruebas, esta metodología fue elegida porque ya existe una primera versión de este proyecto que tuvo como resultado un software de escritorio y lo que se buscaba era en base a este desarrollar los nuevos módulos web que tendrán en cierta medida las mismas funcionalidades, es decir ya se conocían los requerimientos de los nuevos módulos, por esta razón se empleó esta metodología.

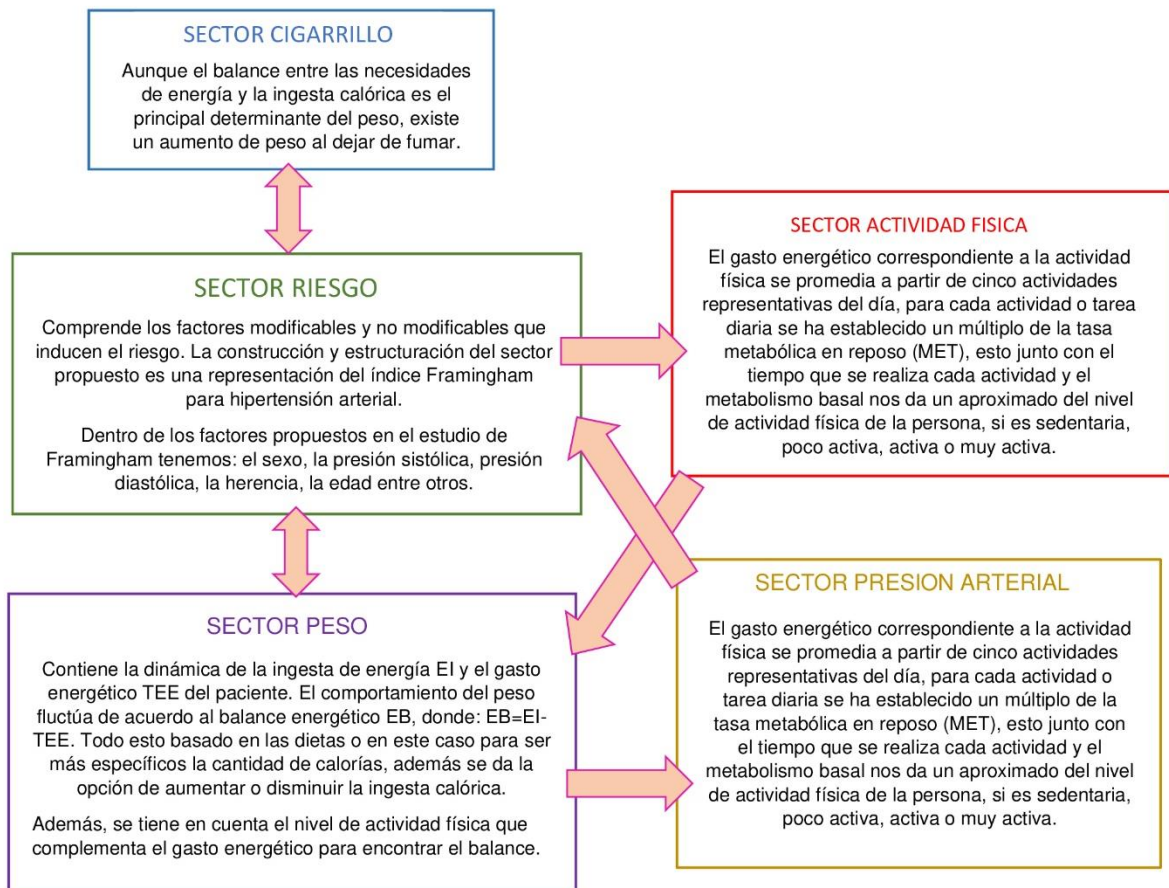
Durante la elaboración del proyecto se realizó la respectiva documentación y así mismo se enuncian propuestas de mejoras para realizar el mantenimiento adecuado del software. Analizando los estudios de este padecimiento social, será de gran ayuda profundizar sobre este problema para dar un enfoque prospectivo, y aportar a la disminución del riesgo de padecer HTA evaluando sus factores principales.

## 6. EVALUACIÓN PRIMERA VERSIÓN

### 6.1 REVISIÓN DE MODELO EVOLUCIÓN

A continuación, se presenta el diagrama por sectores que integran el modelo (figura 2) y el diagrama de flujo nivel respectivo construido con dinámica de sistemas capturados desde el software de modelamiento evolución, cada sector representa un factor que se tuvo en cuenta a la hora de modelar toda la tendencia del riesgo, no se muestra específicamente factores como el riesgo o la edad, ya que están incluidos dentro de los mismos sectores y no requieren de otros cálculos, además se pueden observar en la imagen del modelo.

Figura 2. Mapa de sectores del modelo final





cumpliendo, aun así, todas las pruebas realizadas funcionan de manera correcta. Es importante tener en cuenta que el sector peso del modelo está diseñado para mostrar la tendencia del peso del paciente según su ingesta y su peso ideal, es decir no se hace un cálculo exacto del peso, sino que, basados en la ingesta y en el estado actual, el paciente está en sobre peso, por debajo del peso normal o en el peso ideal y depende sobre todo de la ingesta de calorías y el índice de masa corporal.

Igualmente, el diseño de este proyecto haciendo referencia a la proyección de la enfermedad (modelo) depende en gran medida del comportamiento humano que a su vez está determinado por el entorno en el que esta la persona, lo que hace que el cálculo tenga un margen de error razonable.

Todas las pruebas realizadas se encuentran en el informe pruebas del modelo que se encuentra como Anexo A.




## **6.2 PRUEBAS TÉCNICAS VERSIÓN 1.0**

Se verifico el funcionamiento adecuado del software RIESGO HTA versión 1.0 en diferentes sistemas operativos y así mismo se comprobó su correcto funcionamiento.

Utilizando la mecánica de determinar los sistemas operativos el cual se trabajarán como prueba indicando su funcionalidad en estos y así poder capturar la ejecución del programa de este.

En los siguientes sistemas operativos se realizaron las pruebas de instalación del software RIESGO HTA versión 1.0.

Figura 4. Pruebas de instalación en los distintos SO

Sistema Operativo	Validación
Windows 7 Pro (x64) Windows 8.1 (x64) Windows 10 Home Single Language (x64)	
Apple Mac OS X	
Linux	

La instalación del software de RiesgoHTA 1.0 se hizo en tres (3) pasos, y se analizó su funcionamiento en los sistemas operativos mencionados.

Con esto se verificó y se comprobó que el software RiesgoHTA versión 1.0 funciona en Windows debido a que al ser desarrollado en Delphi 7 tiene una extensión .exe; para los otros sistemas operativos Mac (Apple) que manejan las extensiones .dmg, .pkg, y .iso, y para Linux las extensiones son .dev, .gar, y .gz utilizan una herramienta de compilación Delphi llamada Kylix; el instalador esta comprimido ya que tiene los archivos y al extraerlo se aceptan sus licencias con términos y condiciones, luego de la instalación se crea un acceso directo en el escritorio del software.

La evidencia fotográfica se encuentra en el informe de pruebas software, que se ubica en el Anexo B y en el Anexo C, además incluye una revisión general a todo el software para identificar posibles errores.

### 6.3 PRUEBAS DE RECEPTIVIDAD

Uno de los principales interrogantes que surgió con este proyecto fueron las herramientas sobre las cuales se debía desarrollar el módulo del usuario y el visitador, con la intención de conocer si era viable desarrollar sobre la plataforma

existente o trasladarlo a un ambiente web, definir el diseño y requerimientos de la versión 2 se realizó una consulta sobre la plataforma implementada en la versión 1, las herramientas que la soportan y sobre todo el uso que el ambiente software va a tener, además se tiene en cuenta el resultado de las pruebas de receptividad que se presentan en el anexo D, ya que es importante conocer como los usuarios se relacionan con el software de escritorio de la primera versión 3 años después de su desarrollo.

Algunas pruebas de receptividad fueron aplicadas en la primera versión al final de su desarrollo, sin embargo, teniendo en cuenta el tiempo que ha pasado desde que se hicieron las pruebas (3 años) y el momento de definir los requerimientos de la segunda versión vale la pena obtener otra versión de estas pruebas, puesto que la tecnología avanza a grandes pasos.

Para las pruebas de receptividad se usa el mismo tipo de prueba que en la primera versión pues lo que se quiere es una visión más actual de esta, esta fue realizada con una encuesta escrita, que fue completada por cada médico. Para las pruebas actuales se diseñó una prueba de validación virtual, donde se entregó al médico el manual de usuario, el software, y una encuesta, además teniendo en cuenta que para la segunda versión se plantea tener un diseño y funcionamiento muy similar para el usuario paciente y el visitador, esta prueba se realizó a médicos y enfermeros, con el fin de tener una visión distinta a la de un médico para el módulo del visitador, que podría ser desde un enfermero, un nutricionista o incluso un fisioterapeuta. Las pruebas se pueden encontrar en el Anexo D.

## 6.4 CONCLUSIONES DE LA EVALUACIÓN

Basadas en las pruebas mencionadas anteriormente se concluyó:

1. Implementar en el módulo del paciente y el visitador el mismo modelo que se usó en la primera versión con los respectivos ajustes, puesto que Framighan es el único estudio sustentado, y el cambio del modelo estaría basado en aspectos muy particulares para la entidad interesada, en este caso UISALUD, con ayuda de profesionales de la misma se podría plantear un modelo distinto, dicho esto, en esta versión no se tiene en cuenta ya que se enfoca más en el despliegue de los nuevos módulos y en probar que se podría dar una integración con el sistema de información de la entidad y de esta manera en un proyecto futuro ajustar todo el proyecto a las necesidades de la entidad.
2. Dejar el software de escritorio (versión 1) en la plataforma y estado que se encuentra actualmente, ya que no se considera necesario realizar un mantenimiento a esta, teniendo en cuenta la plataforma en la que está desarrollada, la receptividad que tuvieron los usuarios hacia esta y además que dentro de los nuevos módulos se involucran funciones muy similares, por esto se propone que en una nueva versión se haga la respectiva integración con el módulo del médico.
3. Desarrollar el módulo del paciente y del visitador en plataforma web.
4. El módulo del paciente y del visitador experimentan con un solo escenario (entiéndase por escenario todos los datos requeridos para el cálculo del riesgo de hipertensión sobre todo los que son modificables) donde los datos

serán ingresados por ellos, distinto a la versión del médico donde se plantea un escenario base y un escenario acuerdo, teniendo en cuenta que se busca generar más consciencia en el paciente de su estado de salud por medio de la interfaz web y por su parte el visitador genere una discusión de salud con durante la visita.

## **7. DISEÑO**

El prototipo diseñado en este trabajo de grado está enfocado al servicio de los pacientes y al servicio de los profesionales de la salud, de forma que sirva como apoyo para hacer un seguimiento más estricto a la salud del paciente, no solo de parte de la entidad prestadora del servicio, si no desde el mismo paciente generar una consciencia de cómo sus hábitos diarios pueden impactar su tendencia a sufrir de hipertensión, y la importancia de generar buenos hábitos, en ese sentido este al ser un prototipo intenta facilitar tanto al paciente como al visitador familiar operar el modelo que representa la dinámica del riesgo de hipertensión a través de los años sin necesidad de manejarlo directamente, o sin conocer como está compuesto, solo realizar simulaciones y obtener gráficas a partir de datos ingresados, por otro lado en el módulo del visitador recolectar información que pueda ser útil para la entidad de salud y más adelante poder formar una herramienta más completa que no solo implemente el modelo en la versión actual, si no que permita mostrar una tendencia más acertada e incluso mostrar tendencias relacionadas con otras enfermedades.

Basadas en las conclusiones obtenidas de las pruebas mencionadas en el capítulo 6, es prudente mencionar algunos conceptos con los que se trabajó, además definir las herramientas en las cuales se desarrolló el módulo del paciente y el visitador, en cada sección se ira detallando el desarrollo.

### **7.1 DESCRIPCIÓN GENERAL**

Una cosa importante por resaltar en este trabajo de grado es que el proyecto de investigación dentro del cual está enmarcado pretende crear una forma de atención primaria eficaz, no solo con el desarrollo de software que de alguna manera muestre el riesgo de la salud, si no con el seguimiento del paciente a través de las visitas del profesional de la salud, y de alguna herramienta que alerte a la entidad de salud,

sin embargo, esto requiere un trabajo elaborado con más tiempo y con profesionales de otras áreas.

En esta sección del documento se encuentra la información en relación a las necesidades y características que surgieron del análisis de las pruebas realizadas y que son soportados por los nuevos módulos desarrollados, a pesar de que UISALUD es la entidad interesada y se busca que más adelante el proyecto se ajuste a las necesidades de ellos; para plantear el desarrollo se actuó como cliente y desarrollador, en el momento que finalizó la etapa de desarrollo del prototipo se presentó a la entidad interesada.

La función básica de estos módulos desarrollados es similar a la versión de escritorio, con la diferencia que en este caso se presenta de manera web y como una herramienta para que el usuario (paciente) pueda operarla desde casa o cualquier sitio. Como se mencionó en secciones anteriores se hicieron las evaluaciones pertinentes y se analizó si de alguna manera se podría trabajar sobre esa versión o añadir en el mismo los módulos del paciente y el visitador, finalmente se determinó que la mejor opción es mantener la primera versión en su forma original, pues el principal inconveniente estuvo relacionado con la plataforma en la que se encuentra desarrollada: una versión remota de Delphi cuya licencia había sido adquirida por el grupo de investigación la cual no ha sido renovada, y el hecho de que solo funcione en máquinas con sistema operativo Windows presenta un inconveniente para los usuarios ya que manifestaron que prefieren una interfaz web que pueda ser accedida desde cualquier máquina independiente del sistema operativo y no necesite una instalación previa, si no que su acceso sea instantáneo, considerando esto se plantea como recomendación futura añadir el módulo del médico a la última versión (módulo del paciente y del visitador) si la entidad interesada lo solicita o se considera pertinente. Además, para esta versión no se pretende cambiar el modelo drásticamente, se verificó que funciona correctamente y se implementó en la interfaz web.

Siguiendo esta idea los módulos a desarrollar tendrán como función:

- Crear, modificar pacientes
- Registrar y mostrar las visitas de los profesionales de la salud
- Hacer simulaciones de la tendencia a sufrir de hipertensión arterial

## **7.2 MODELO EVOLUCIÓN**

Teniendo en cuenta que no existe un estudio justificado distinto al de Framingham donde se incluyan nuevos factores que puedan ayudar a predecir de manera más exacta el riesgo de hipertensión y en este momento un nuevo modelo que se ajuste a la entidad interesada debe ser en parte diseñado por ellos y bajo las condiciones que consideren importantes, teniendo en cuenta esto el modelo debe mantener los mismos sectores y se hicieron dos ajustes basados en las pruebas que se encuentran en el anexo A, como se diseñó el módulo del paciente y del visitador el modelo genera la gráfica de un escenario y el usuario podrá modificar los valores y generar nuevamente la gráfica, en este sentido los módulos además de ser un soporte para la entidad de salud lo son para el paciente, el mismo puede interactuar con la interfaz generar graficas e interpretar resultados debido a que la alimentación, en este caso la cantidad de calorías y la actividad física son unos de los principales factores en el riesgo de la hipertensión y tiene mucho que ver con el ambiente familiar, ya que es complejo que una persona lleve una buena alimentación si su entorno no coopera, es por esto que se plantea como una estrategia de atención primaria no solo desde la entidad de salud si no desde el hogar.

## **7.3 FUNCIONALIDADES**

La segunda versión planteada es un prototipo de lo que sería HTA 2.0 desarrollado en un ambiente web con todos los módulos necesarios, para hacer la descripción

de las funcionalidades del software se presentan a continuación las plantillas de las técnicas usadas y la métrica que las define:

Tabla 1. Plantilla para la definición de requerimientos funcionales

<b>ID</b>	<Identificador del requerimiento>		
<b>Nombre</b>	<Nombre representativo que define el requerimiento>		
<b>Complejidad</b>	<Que tan difícil es cumplir con el requerimiento>	<b>Prioridad</b>	<Prioridad del requerimiento de acuerdo a la métrica (1.5.4) >
<b>Usuarios</b>	<Quien hará uso de la aplicación>		
<b>Descripción</b>			
<Descripción resumida y clara del requerimiento>			
<b>Consideraciones</b>			
<Descripción más detallada del requerimiento y lo que se debe tener en cuenta para su completo desarrollo>			

Tabla 2. Plantilla de requerimientos No funcionales

<b>Identificación del requerimiento:</b>	<Identificador del requerimiento>	<b>Prioridad:</b>	<Prioridad del requerimiento de acuerdo a la métrica (1.5.4) >
<b>Nombre</b>	<Nombre representativo que define el requerimiento>		
<b>Descripción del requerimiento:</b>	<Descripción detallada y clara del requerimiento>		

**7.3.1 Especificación de requerimientos.** Ya teniendo el modelo básico a emplear, se dio inicio a sacar las funcionalidades y se definen de forma específica.

### 7.3.1.1 Requerimientos funcionales

Tabla 3. RF01-Registrar paciente

<b>ID</b>	01		
<b>Nombre</b>	Registrar paciente		
<b>Complejidad</b>	Media	<b>Prioridad</b>	5
<b>Usuarios</b>	Pacientes		

<b>Descripción</b>		
<i>El paciente podrá entrar en la plataforma y hacer clic sobre un botón ¿no estas registrado?, donde pondrá los datos necesarios incluida una contraseña para su inicio de sesión posterior.</i>		
<b>Consideraciones</b>		
<i>Ninguna</i>		

Tabla 4. RF02-Iniciar sesión por roles

<b>ID</b>	02		
<b>Nombre</b>	<i>Iniciar sesión por roles</i>		
<b>Complejidad</b>	<i>Media</i>	<b>Prioridad</b>	5
<b>Usuarios</b>	<i>Pacientes y visitantes</i>		
<b>Descripción</b>			
<i>El paciente y los visitantes pueden iniciar sesión de acuerdo con su rol a través de su respectiva cédula y la contraseña que fue guardada en el registro.</i>			
<b>Consideraciones</b>			
<i>Los visitantes no se registran en la página, por lo tanto su registro lo hace directamente el administrador del sistema, su usuario para el inicio de sesión es igual que el de los pacientes, la cédula, y la contraseña será asignada por el administrador también.</i>			

Tabla 5. RF03-Hacer simulaciones

<b>ID</b>	03		
<b>Nombre</b>	<i>Hacer simulaciones</i>		
<b>Complejidad</b>	<i>Media</i>	<b>Prioridad</b>	5
<b>Usuarios</b>	<i>Pacientes y visitantes</i>		
<b>Descripción</b>			
<i>Los pacientes y los visitantes podrán realizar simulaciones con los datos necesarios cuantas veces quieran, sin embargo, estas no se guardan, solo es un estímulo para que el paciente juegue con su alimentación y actividad física, y de la misma manera el visitante pueda mostrarle al paciente como se comportara el riesgo en el futuro de acuerdo a sus condiciones del momento.</i>			
<i>La simulación hace referencia a una gráfica que se genera en base a la información que el usuario le suministre y muestra el riesgo que tiene el paciente de sufrir hipertensión en los siguientes años.</i>			
<b>Consideraciones</b>			
<i>Algunos de los datos necesarios se guardarán, de manera que el paciente o el visitante no deban escribir datos innecesariamente</i>			

Tabla 6. RF04-Registrar visitas

<b>ID</b>	04		
<b>Nombre</b>	Registrar visitas		
<b>Complejidad</b>	Media	<b>Prioridad</b>	5
<b>Usuarios</b>	Visitador		
<b>Descripción</b>			
<p>Los visitantes son la relación que de alguna manera existe entre el médico y el paciente, se plantea que el visitador haga visitas al paciente para revisar su estado, el visitador después de iniciar sesión podrá hacer clic sobre un botón donde se abrirá el formulario que inicialmente tendrá la misma simulación que hace el paciente, asimismo buscará que se cree un dialogo en base a este, luego se despliegan los demás campos del formulario y el visitador registra la información necesaria relacionada a la visita.</p>			
<b>Consideraciones</b>			
Ninguna			

Tabla 7. RF05-Editar paciente

<b>ID</b>	05		
<b>Nombre</b>	Editar Paciente		
<b>Complejidad</b>	Media	<b>Prioridad</b>	5
<b>Usuarios</b>	Paciente		
<b>Descripción</b>			
<p>El paciente podrá editar su información personal tal como nombre, apellidos, número de documento, correo electrónico, contraseña, celular ciudad, dirección y barrio, pero será lo único a lo que tendrá acceso y podrá modificar, teniendo en cuenta que si cambia su lugar de residencia o su número de celular pueda actualizarlo y así la entidad interesada tener esta información al día</p>			
<b>Consideraciones</b>			
Ninguna			

Tabla 8. RF06-Listar pacientes

<b>ID</b>	06		
<b>Nombre</b>	Listar Paciente		
<b>Complejidad</b>	Media	<b>Prioridad</b>	5
<b>Usuarios</b>	Visitador		
<b>Descripción</b>			
<p>Cada vez que el visitador inicia sesión se despliega una lista de los pacientes que ha visitado, sin embargo, si el paciente no está en su lista, el visitador tiene acceso a cualquier paciente que desee, y agregarle una visita.</p>			
<b>Consideraciones</b>			

<i>Ninguna</i>
----------------

Tabla 9. RF07-Historial de visitas

<b>ID</b>	<i>07</i>		
<b>Nombre</b>	<i>Historial</i>		
<b>Complejidad</b>	<i>Media</i>	<b>Prioridad</b>	<i>5</i>
<b>Usuarios</b>	<i>Pacientes, Visitador</i>		
<b>Descripción</b>	<i>Tanto el paciente como el visitador tendrá acceso al historial de visitas que se han realizado, es decir el paciente podrá ver su historial de visitas en su rol de paciente, y el visitador desde su módulo, podrá ver el historial de todos los pacientes que pertenecen a su lista.</i>		
<b>Consideraciones</b>	<i>Ninguna</i>		

### 7.3.1.2 Requerimientos NO funcionales

Tabla 10. RNF01 - Interfaz de usuario

<b>Identificación del requerimiento:</b>	<i>RNF01</i>	<b>Prioridad del requerimiento:</b>	<i>5</i>
<b>Nombre:</b>	<i>Interfaz del usuario</i>		
<b>Descripción del requerimiento:</b>	<p><i>Los módulos deben ser desarrollados para ser ejecutado en un ambiente web, debe ser amigable e intuitiva para el usuario, el diseño se caracteriza por ser “Responsive” a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes. Además, el sistema debe permitir su navegación a través de los exploradores más comunes como Mozilla Firefox, Internet Explorer, Chrome y las diferentes plataformas (Windows, Mac, Linux).</i></p> <p><i>El navegador no debe requerir ninguna modificación o instalación de plugins, applets, o similares para que el software funcione, ni requerir soporte técnico al usuario para poder operar la aplicación.</i></p>		

Tabla 11. RNF02 - Seguridad

<b>Identificación del requerimiento:</b>	<i>RNF02</i>	<b>Prioridad:</b>	<i>5</i>
<b>Nombre</b>	<i>Seguridad</i>		

<b>Descripción del requerimiento:</b>	<i>Se deben cumplir con los debidos protocolos de seguridad para bases de datos, que la información sea clara y verídica, además los módulos deben contar con la seguridad necesaria para el inicio de sesión, y el manejo de datos de los pacientes, y teniendo en cuenta que esto será montado en un servidor del grupo de investigación SIMON, este debe contar con las especificaciones mínimas de seguridad.</i>
---------------------------------------	---

Tabla 12. RNF03 - Disponibilidad

<b>Identificación del requerimiento:</b>	<i>RNF03</i>	<b>Prioridad:</b>	5
<b>Nombre</b>	<i>Disponibilidad</i>		
<b>Descripción del requerimiento:</b>	<i>La disponibilidad de los módulos deberá ser permanente pero también dependerá de la disponibilidad del proveedor de acceso a Internet o de los servicios de interconexión prestados por terceros.</i>		

Tabla 13. RNF05 - Desempeño

<b>Identificación del requerimiento:</b>	<i>RNF04</i>	<b>Prioridad:</b>	5
<b>Nombre</b>	<i>Desempeño</i>		
<b>Descripción del requerimiento:</b>	<p><i>Los tiempos de respuesta relacionados al manejo de formularios, en forma general, no debe ser superior a 3 segundos.</i></p> <p><i>El despliegue de informes y consultas dependiendo de su complejidad media, se espera que no exceda un tiempo de 5 segundos, sin embargo, teniendo en cuenta la simulación quizá se muestre en tiempo real esta debe tomar un tiempo prudente pero que no sobrepase el minuto de respuesta</i></p> <p><i>Lo anterior se debe poder obtener en un ambiente tecnológico controlado, que permita tener recursos de procesamiento, almacenamiento y comunicaciones disponibles solo para el sistema.</i></p>		

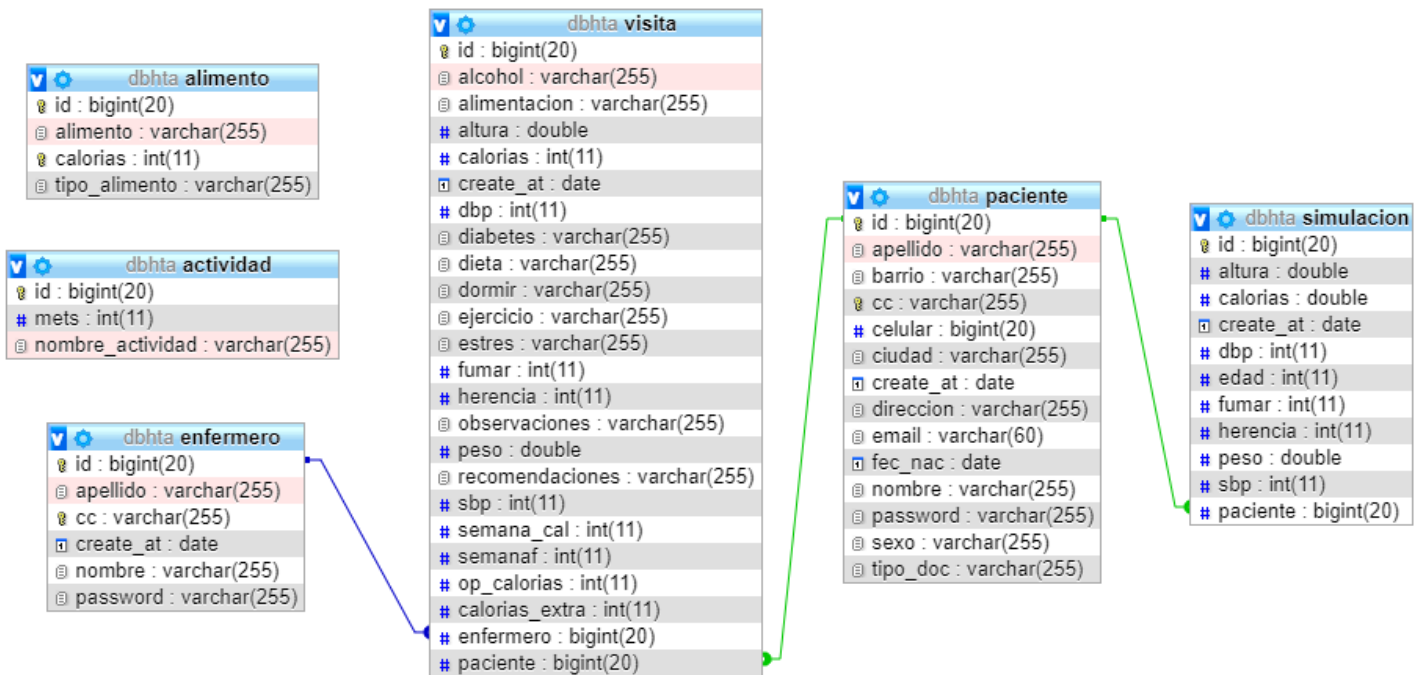
## 7.4 DIAGRAMAS UML

A continuación, se presenta de forma general el diseño UML de este proyecto con los diagramas que se consideran pertinentes.

**7.4.1 Diagrama entidad relación.** En este prototipo se implementó una base de datos parcial que busca almacenar tanto a los pacientes, como a los visitantes, y más allá de guardar la simulación, se busca almacenar la información que pueda ser relevante para la entidad, a diferencia de la versión de escritorio que almacena toda la información del escenario acuerdo y del escenario base.

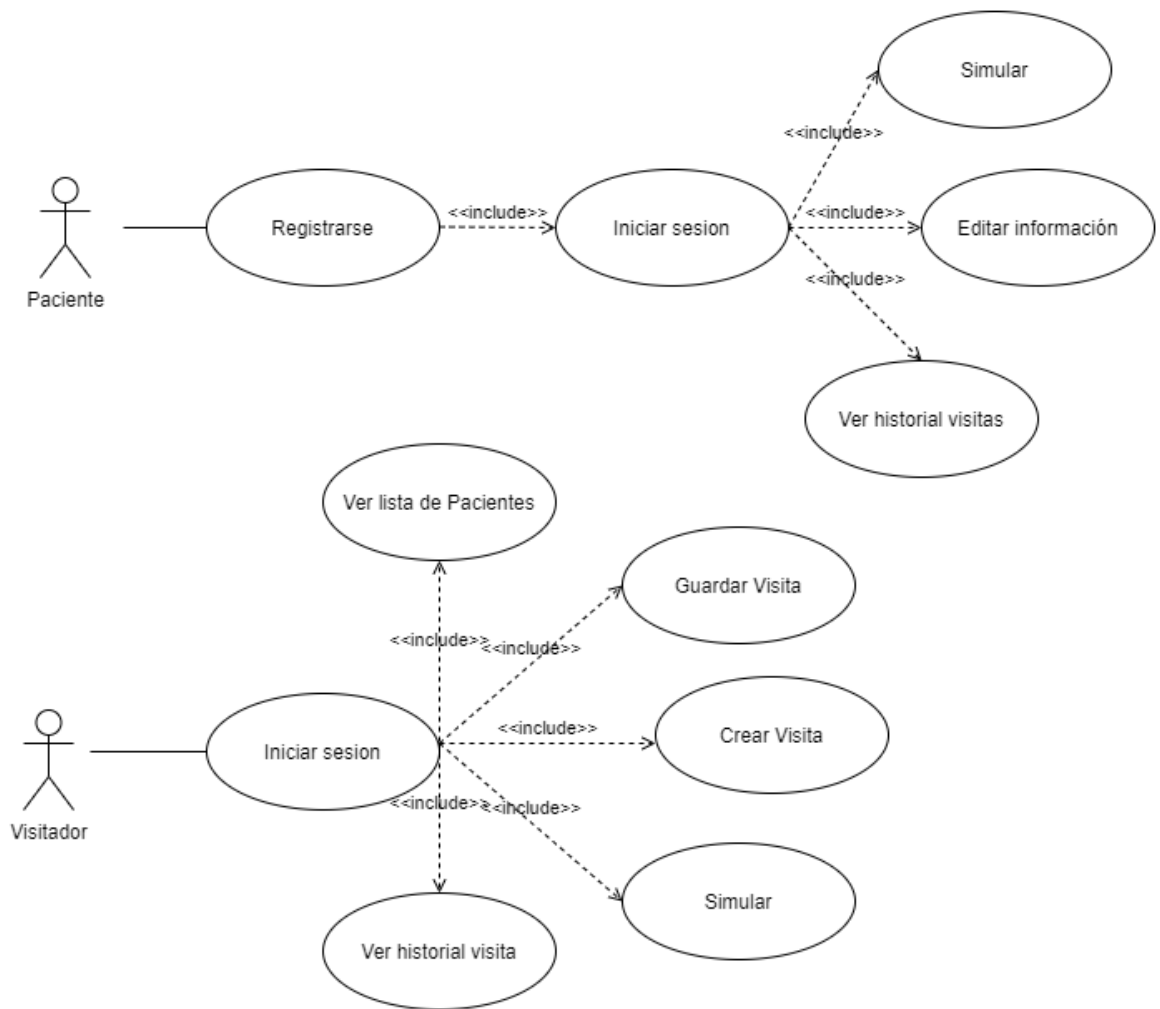
Hay que tener en cuenta que el módulo del paciente y del visitador están separados, es decir el módulo paciente está orientando a que cualquier persona lo usé sea o no paciente de UISALUD u otra entidad de salud, con el fin de que puedan operar con el modelo y obtener resultados, mientras que el módulo del visitador estará diseñado para que solo profesionales de la salud puedan operarlo y guardar la información, ya que el fin es que por medio de la simulación el visitador pueda discutir con el paciente y el entorno familiar, y guardar la información que pueda ser de interés para controlar la salud del paciente.

Figura 5. Diagrama entidad relación



**7.4.2 Diagrama de casos de uso.** A continuación, se presenta un diagrama general de casos de uso que integra todas las funcionalidades, un diagrama de actividades generales y después se hará la descripción de cada uno de los casos de uso.

Figura 6. Diagrama de caso de usos



### 7.4.3 Diagrama de actividades

Figura 7. Diagrama de actividades del paciente

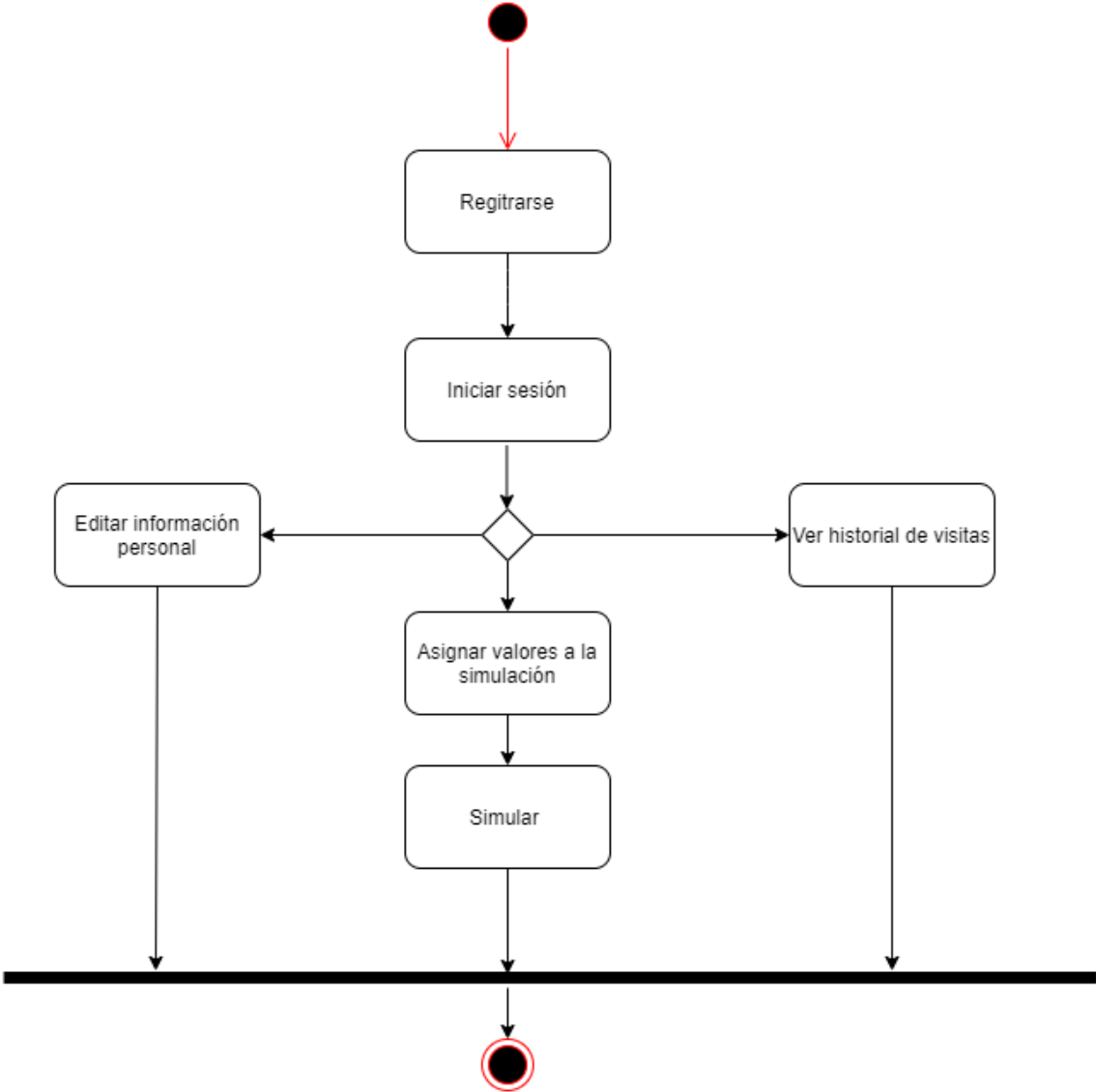
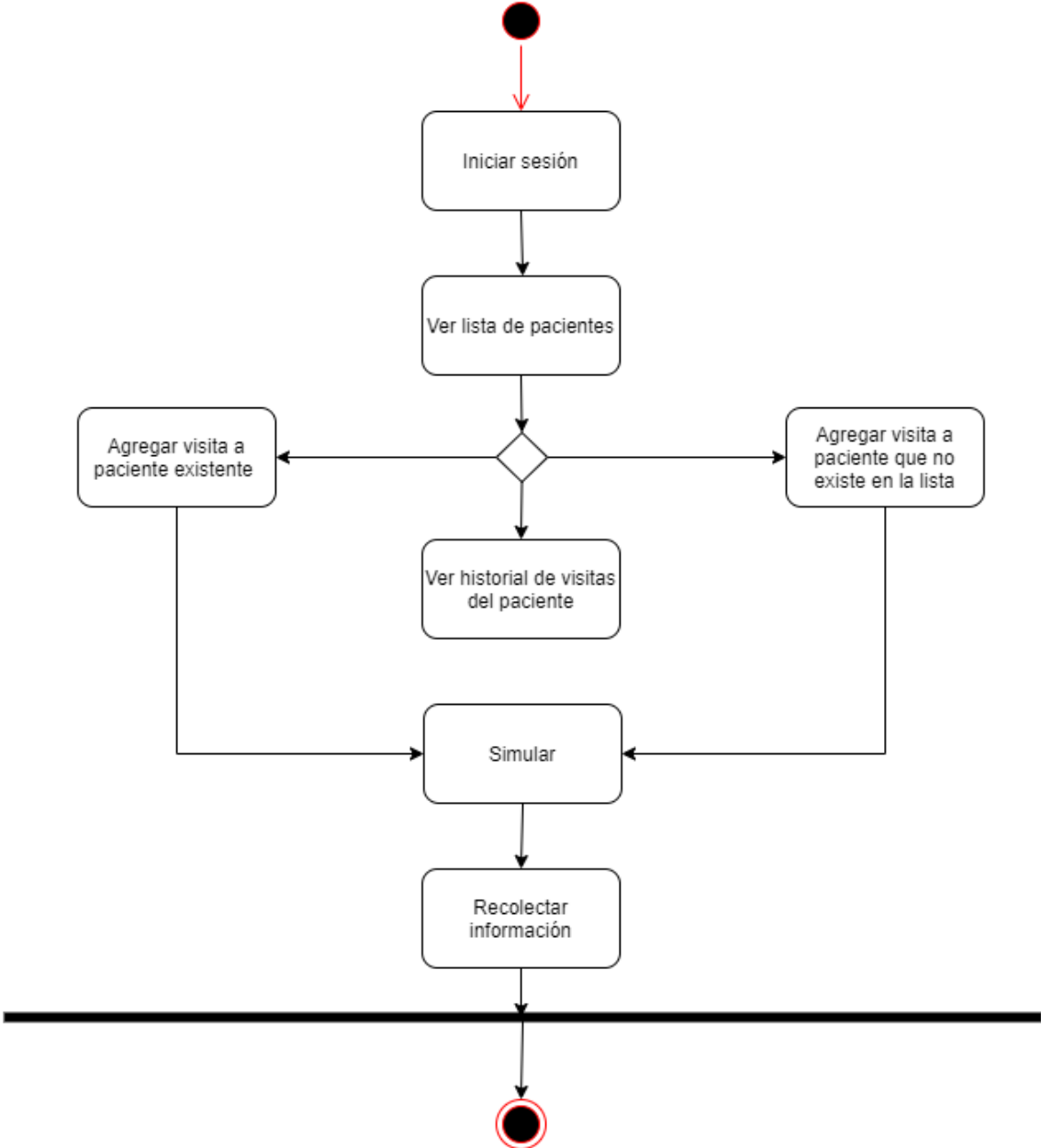


Figura 8. Diagrama de actividades del visitador



## 7.5 DESCRIPCIÓN CASOS DE USO

Ahora vamos a presentar los casos de uso que se presentan en base a los requerimientos descritos en la sección 7.3.1.1, pero en esta sección se hace una descripción más detallada de estos.

Esta es la plantilla para describir los casos de uso:

Tabla 14. Plantilla casos de uso

<b>ID:</b>		<b>Id Requerimiento:</b>	
<b>Nombre:</b>			
<b>Actores:</b>			
<b>Descripción</b>			
<b>Pre-condiciones</b>			
<b>Flujo normal</b>			
<b>Flujos alternativos</b>			
<b>Post-condiciones</b>			

Tabla 15. CU1 Iniciar sesión

<b>ID:</b>	<i>CUI</i>	<b>Id Requerimiento:</b>	<i>RF02</i>
<b>Nombre:</b>	<i>Iniciar sesión</i>		
<b>Actores:</b>	<i>Paciente y visitador</i>		
<b>Descripción</b>			
<i>El caso de uso inicia cuando el paciente o visitador ingresa a la plataforma de RiesgoHta2.0 para iniciar sesión.</i>			

<b>Pre-condiciones</b>
<p><i>El paciente debe registrarse primero en la plataforma para poder iniciar sesión en caso de que sea paciente de la entidad de salud el usuario y la contraseña serán su cédula.</i></p> <p><i>El visitador debe estar registrado previamente, directamente en la base de datos, debe usar el usuario y la contraseña que le fueron asignados.</i></p>
<b>Flujo normal</b>
<ol style="list-style-type: none"> <li>1. <i>El usuario ingresa a la plataforma de HTA</i></li> <li>2. <i>El usuario elige su rol (visitador o paciente)</i></li> <li>3. <i>El usuario pone su usuario y contraseña</i></li> <li>4. <i>El usuario hace clic sobre iniciar sesión</i></li> <li>5. <i>El usuario inicio sesión</i></li> </ol>
<b>Flujos alternativos</b>
<p><b>EL USUARIO PONE SUS DATOS INCORRECTOS</b></p> <ol style="list-style-type: none"> <li>1. <i>El usuario ingresa a la plataforma de HTA</i></li> <li>2. <i>El usuario elige su rol (visitador o paciente)</i></li> <li>3. <i>El usuario pone su usuario y contraseña</i></li> <li>4. <i>El usuario hace clic sobre iniciar sesión</i></li> <li>5. <i>Se muestra un error porque el rol, usuario o contraseña son incorrectos</i></li> <li>6. <i>El usuario pone sus datos corregidos y hace clic sobre entrar</i></li> <li>7. <i>El usuario inicio sesión</i></li> </ol> <p><b>EL USUARIO NO ESTA REGISTRADO</b></p> <ol style="list-style-type: none"> <li>1. <i>El usuario ingresa a la plataforma de HTA</i></li> <li>2. <i>El usuario elige su rol (visitador o paciente)</i></li> <li>3. <i>El usuario pone su usuario y contraseña</i></li> <li>4. <i>El usuario hace clic sobre iniciar sesión</i></li> <li>5. <i>Se muestra un mensaje de que el usuario o contraseña no existe</i></li> <li>6. <i>El paciente se registra</i></li> </ol>

5. <i>El paciente inicia sesión</i>
<b>Post-condiciones</b>
<i>Ninguna</i>

Tabla 16. CU2 Registrarse

<b>ID:</b>	CU2	<b>Id Requerimiento:</b>	<i>RF01</i>
<b>Nombre:</b>	<i>Registrarse</i>		
<b>Actores:</b>	<i>Paciente</i>		
<b>Descripción</b>			
<i>El caso de uso inicia cuando el paciente que no pertenece a la entidad de salud quiere usar la plataforma, pero no se ha registrado aún.</i>			
<b>Pre-condiciones</b>			
<i>Ninguna</i>			
<b>Flujo normal</b>			
<ol style="list-style-type: none"> <li>1. <i>El paciente hace clic sobre el botón: ¿no estas registrado?</i></li> <li>2. <i>El paciente pone los datos correspondientes en el formulario</i> <ul style="list-style-type: none"> <li>• <i>Nombres</i></li> <li>• <i>Apellidos</i></li> <li>• <i>Sexo</i></li> <li>• <i>Fecha de nacimiento</i></li> <li>• <i>Tipo de documento</i></li> <li>• <i>Número de documento</i></li> <li>• <i>Correo electrónico</i></li> <li>• <i>Contraseña</i></li> <li>• <i>Celular o teléfono</i></li> <li>• <i>Ciudad</i></li> <li>• <i>Dirección</i></li> <li>• <i>Barrio</i></li> </ul> </li> <li>3. <i>El paciente hace clic sobre guardar</i></li> </ol>			

4. <i>La plataforma vuelve a la página principal de inicio de sesión</i>
<b>Flujos alternativos</b>
<p><b><i>EL PACIENTE VA AL INICIO DE SESIÓN ANTES DE GUARDAR</i></b></p> <ol style="list-style-type: none"> <li>1. <i>El paciente se está registrando y antes de guardar intenta ir al inicio</i></li> <li>2. <i>Se muestra un mensaje de alerta, preguntando si está seguro de salir y perder los cambios:</i> <ol style="list-style-type: none"> <li>2.1 <i>El paciente hace clic en aceptar</i> <ol style="list-style-type: none"> <li>2.1.1 <i>La página se redirige al inicio de sesión, pero el paciente no puede operar</i></li> <li>2.2 <i>El paciente hace clic sobre cancelar</i> <ol style="list-style-type: none"> <li>2.2.1 <i>Se mantiene en el formulario y termina de llenar la información</i></li> <li>2.2.2 <i>Hace clic sobre guardar</i></li> <li>2.2.3 <i>Vuelve al inicio de sesión</i></li> </ol> </li> </ol> </li> </ol> </li> </ol> <p><b><i>EL PACIENTE DEJA CAMPOS VACIOS</i></b></p> <ol style="list-style-type: none"> <li>1. <i>El paciente está llenando su registro y deja campos vacíos</i></li> <li>2. <i>El paciente hace clic en guardar</i></li> <li>3. <i>Se muestra un mensaje indicando que hay campos vacíos</i></li> <li>4. <i>Hasta que el formulario se llene por completo no se puede guardar</i></li> </ol>
<b>Post-condiciones</b>
<i>Ninguna</i>

Tabla 17. CU3 Editar paciente

<b>ID:</b>	CU3	<b>Id Requerimiento:</b>	RF05
<b>Nombre:</b>	<i>Editar paciente</i>		
<b>Actores:</b>	<i>Paciente</i>		
<b>Descripción</b>			

<i>El caso de uso se da cuando el paciente necesita o quiere actualizar sus datos personales.</i>
<b>Pre-condiciones</b>
<i>El paciente debe estar registrado en la plataforma para que sus datos sean modificados.</i>
<b>Flujo normal</b>
<ol style="list-style-type: none"> <li>1. <i>El paciente hace clic sobre el icono de configuración que estará en la parte derecha superior</i></li> <li>2. <i>Hace clic sobre editar información</i></li> <li>3. <i>Se abre el formulario con los siguientes datos del paciente:</i> <ul style="list-style-type: none"> <li>• <i>Nombres</i></li> <li>• <i>Apellidos</i></li> <li>• <i>Sexo</i></li> <li>• <i>Número de documento</i></li> <li>• <i>Correo electrónico</i></li> <li>• <i>Contraseña</i></li> <li>• <i>Celular</i></li> <li>• <i>Ciudad</i></li> <li>• <i>Dirección</i></li> <li>• <i>Barrio</i></li> </ul> </li> <li>4. <i>El paciente modifica los datos que desea actualizar</i></li> <li>5. <i>El paciente hace clic sobre actualizar</i></li> <li>6. <i>Se actualizan los campos</i></li> </ol>
<b>Flujos alternativos</b>
<b><i>EL PACIENTE BORRA UN CAMPO, PERO LO DEJA VACÍO</i></b>
<ol style="list-style-type: none"> <li>1. <i>El paciente inicia sesión</i></li> <li>2. <i>El paciente hace clic sobre actualizar datos</i></li> <li>3. <i>Se abre el formulario con los datos del paciente</i></li> <li>4. <i>El paciente borra un dato, pero olvida actualizarlo</i></li> </ol>

5. *El paciente hace clic sobre guardar*
6. *Se muestra un mensaje de alerta, de que los campos están vacíos*
7. *El paciente modifica todos los campos o los deja iguales*
8. *Se actualizan los datos*
9. *Vuelve a la simulación del paciente*

**EL PACIENTE VA AL INICIO ANTES DE GUARDAR**

1. *El paciente se está modificando sus datos y antes de guardar la actualización intenta ir al inicio*
2. *Se muestra un mensaje de alerta, preguntando si está seguro de salir y perder los cambios:*
  - 2.1 *El paciente hace clic en aceptar*
    - 2.1.1 *La página se redirige a la simulación*
  - 2.2 *El paciente hace clic sobre cancelar*
    - 2.2.1 *Se mantiene en el formulario termina de llenar la información*
    - 2.2.2 *Hace clic sobre actualizar*
    - 2.2.3 *Vuelve a simular*

**Post-condiciones**

*Ninguna*

Tabla 18. CU4 Agregar visita

<b>ID:</b>	CU4	<b>Id Requerimiento:</b>	RF04
<b>Nombre:</b>	Agregar visita		
<b>Actores:</b>	Visitador		
<b>Descripción</b>			
<p><i>Después de que el visitador inicia sesión, se desplegara la lista de pacientes que él tiene asociados, a continuación, tendrá un botón donde podrá agregar una visita a uno de los pacientes existentes o en caso de no haber realizado una visita previa al paciente agregar uno nuevo y generar la primera visita.</i></p>			

**Pre-condiciones**

*El visitador debe estar logueado*

**Flujo normal**

- 1. El visitador inicia sesión*
- 2. El visitador hace clic sobre el icono de agregar visita correspondiente al paciente que está visitando*
- 3. El visitador digita la información relacionada con la alimentación, la actividad física, la presión entre otros para poder simular.*
- 4. El visitador simula y discute con el paciente.*
- 5. Se despliegan otros campos para recolectar información adicional*
- 6. El visitador llena toda la información*
- 7. El visitador hace clic sobre guardar visita*
- 8. Se guarda la visita*

**Flujos alternativos*****EL PACIENTE NO ESTA EN LA LISTA DE PACIENTES***

- 1. El visitador inicia sesión*
- 2. El paciente al que le va a realizar la visita no está en la lista de pacientes visitados*
- 3. El visitador hace clic sobre el botón: agregar paciente*
- 4. El visitador digita la información relacionada con la alimentación, la actividad física, la presión entre otros para poder simular.*
- 5. El visitador simula y discute con el paciente.*
- 6. Se despliegan otros campos para recolectar información adicional*
- 7. El visitador llena toda la información*
- 8. El visitador hace clic sobre guardar visita*
- 9. Se guarda la visita*

***CAMPOS NULOS EN LOS FORMULARIOS***

- 1. El visitador inicia sesión*
- 2. El visitador hace clic sobre el botón + visita o agregar paciente si el paciente no está en la lista*

<ol style="list-style-type: none"> <li>3. El visitador digita la información relacionada con la alimentación, la actividad física, la presión entre otros para poder simular.</li> <li>4. Se muestra una alerta de campos vacíos</li> <li>5. El visitador completa la información simula y discute con el paciente.</li> <li>6. Se despliegan otros campos para recolectar información adicional</li> <li>7. El visitador llena toda la información</li> <li>6. Aparece una alerta de campos vacíos</li> <li>7. El visitador confirma que todo esté lleno, completa y vuelve a hacer clic sobre guardar</li> </ol>
<b>Post-condiciones</b>
Ninguna

Tabla 19. CU5 Historial de visitas

<b>ID:</b>	CU5	<b>Id Requerimiento:</b>	RF07
<b>Nombre:</b>	<i>Historial</i>		
<b>Actores:</b>	<i>Paciente y visitador</i>		
<b>Descripción</b>			
<i>El caso de uso se da cuando el paciente quiere conocer las visitas que le han realizado o el visitador quiere ver las visitas que le han realizado al paciente</i>			
<b>Pre-condiciones</b>			
<i>Estar logueado</i>			
<b>Flujo normal</b>			
<b>PARA EL PACIENTE:</b>			
<ol style="list-style-type: none"> <li>1. El paciente hace clic sobre el icono de configuración que estará en la parte derecha superior</li> <li>2. Hace clic sobre historial</li> <li>3. Se muestra una tabla con las visitas que le han realizado con la siguiente la información: <ul style="list-style-type: none"> <li>• Fecha de la visita</li> </ul> </li> </ol>			

<ul style="list-style-type: none"> <li>• <i>Nombre del visitador que realizo la visita</i></li> <li>• <i>Presión sistólica – presión diastólica</i></li> <li>• <i>Peso</i></li> <li>• <i>Observaciones</i></li> </ul> <p><b>PARA EL VISITADOR</b></p> <ol style="list-style-type: none"> <li>1. <i>El visitador hace clic sobre el botón de historial que está al frente del paciente</i></li> <li>2. <i>Se muestra una tabla con las visitas que le han realizado al paciente con la siguiente la información:</i> <ul style="list-style-type: none"> <li>• <i>Fecha de la visita</i></li> <li>• <i>Nombre del visitador que realizo la visita</i></li> <li>• <i>Presión sistólica – presión diastólica</i></li> <li>• <i>Peso</i></li> <li>• <i>Observaciones</i></li> </ul> </li> </ol>
<b>Flujos alternativos</b>
<i>Ninguno</i>
<b>Post-condiciones</b>
<i>Ninguna</i>

Tabla 20. CU6 Simular

<b>ID:</b>	CU6	<b>Id Requerimiento:</b>	RF03
<b>Nombre:</b>	<i>Simular</i>		
<b>Actores:</b>	<i>Paciente y visitador</i>		
<b>Descripción</b>			
<i>Tanto el visitador como el paciente podrán hacer la simulación que muestra la tendencia a sufrir de hipertensión arterial, tendrán una especie de formulario donde digitaran la información pertinente sobre la actividad física y la dieta que calculara la tendencia del riesgo el cual se mostrará en una gráfica.</i>			
<b>Pre-condiciones</b>			

<i>Estar logueado</i>
<b>Flujo normal</b>
<p><i>Para el paciente:</i></p> <ol style="list-style-type: none"> <li>1. <i>El paciente llena los datos pertinentes para hacer la simulación</i></li> <li>2. <i>Hace clic sobre: mostrar tendencia</i></li> <li>3. <i>Se muestra la gráfica de:</i> <ul style="list-style-type: none"> <li>• <i>Tendencia a sufrir de hipertensión</i></li> <li>• <i>Peso</i></li> </ul> </li> <li>4. <i>Si el paciente desea cambiar algún dato lo puede hacer y después hace clic sobre el botón actualizar datos que está encima de la gráfica</i></li> <li>5. <i>Se generan las nuevas gráficas con los nuevos datos</i></li> </ol> <p><i>Para el visitador:</i></p> <ol style="list-style-type: none"> <li>1. <i>El visitador crea una visita</i></li> <li>2. <i>Llena los datos necesarios para la simulación</i></li> <li>3. <i>Se muestran las gráficas de:</i> <ul style="list-style-type: none"> <li>• <i>Tendencia al riesgo de hipertensión</i></li> <li>• <i>Peso</i></li> </ul> </li> <li>4. <i>Si el visitador quiere simular con otros datos puede hacerlo, solo debe cambiar los datos y hacer clic sobre el botón actualizar datos que está debajo de la grafica</i></li> </ol>
<b>Flujos alternativos</b>
<i>Ninguno</i>
<b>Post-condiciones</b>
<i>Ninguno</i>

Tabla 21. CU7 Cerrar sesión

<b>ID:</b>	CU7	<b>Id Requerimiento:</b>	RF02
<b>Nombre:</b>	Cerrar sesión		

<b>Actores:</b>	<i>Paciente y visitador</i>
<b>Descripción</b>	
<i>El caso de uso se inicia cuando el usuario desea salir del sistema, culminando de esta forma el caso de uso.</i>	
<b>Pre-condiciones</b>	
<i>El usuario debe estar logueado.</i>	
<b>Flujo normal</b>	
<b><i>PARA EL PACIENTE</i></b>	
<ol style="list-style-type: none"> <li><i>1. El paciente hace clic sobre el icono de configuración</i></li> <li><i>2. El paciente hace clic sobre cerrar sesión</i></li> <li><i>3. Se cierra la sesión</i></li> </ol>	
<b><i>PARA EL VISITADOR</i></b>	
<ol style="list-style-type: none"> <li><i>1. El visitador hace clic sobre cerrar sesión</i></li> <li><i>2. Se cierra la sesión</i></li> </ol>	
<b>Flujos alternativos</b>	
<i>Ninguno</i>	
<b>Post-condiciones</b>	
<i>Ninguna</i>	

## **8. DESARROLLO**

En esta versión se construyó un prototipo de aplicación web el cual es más amigable y accesible, desarrollando en este caso los módulos visitador y paciente, en el caso del visitador, este podrá registrar una visita que realiza al paciente asignado captando información que puede ser relevante para la entidad de salud, de igual manera el usuario paciente puede interactuar con el aplicativo mostrándole una tendencia aproximada de su riesgo de hipertensión arterial basado en su estado actual o con los valores que el paciente decida jugar dándole así más conocimiento y control no solo a él sino también a sus familiares con respecto a su estado de salud.

Para la parte del Back-end se utilizaron los recursos ApiRest, debido a que este estándar es más lógico, eficiente y habitual para servicios de internet, en otras palabras esta interfaz usa HTTP para obtener los datos o generar operaciones sobre los datos recibidos por el cliente y se implementaron frameworks de desarrollo como Spring MVC, específicamente Spring-Boot, todo esto bajo el lenguaje de java y así mismo en relación con el Front-end se empleó Angular utilizando su librería de estilos aplicando una apariencia definida por angular material (Material Design) y se basó en el patrón de diseño modelo – vista – controlador (MVC).

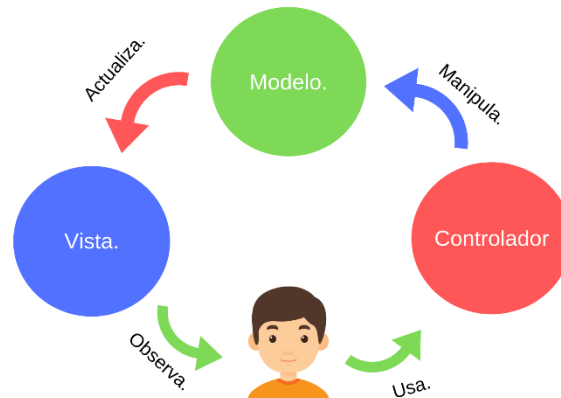
### **8.1 MODELO VISTA CONTROLADOR**

El modelo vista controlador es un modelo de arquitectura de software que separa la lógica de la aplicación (el modelo) de su representación (la vista) y para la interacción entre ambas introduce el controlador que se encarga de trasladar las solicitudes provenientes de la vista a la aplicación para que esta haga lo que corresponda y luego le devuelva el resultado al controlador y que sea este el que le envíe a la vista el resultado para que lo represente.

El objetivo del patrón MVC es separar la vista de la lógica de la aplicación para facilitar la reutilización de código y simplificar el desarrollo de ambas capas de forma separada para facilitar el cambio de la vista o para crear aplicaciones con múltiples vistas (web, android, ios, pdf, entre otros).

Además, teniendo en cuenta las ventajas de separar el front-end del back-end como: simplificación del trabajo, facilitar la reutilización de código, separación de conceptos y hace más sencilla la tarea de su posterior mantenimiento, entre otras y sumado a las ventajas que presenta la arquitectura fue la implementada.

Figura 9. Modelo vista controlador



Fuente: Valeria García Cobián, Patrón de diseño MVC, Disponible en <<https://blog.nearsoftjobs.com/patr%C3%B3n-de-dise%C3%B1o-mvc-2366948b5fc7>>

De acuerdo con la imagen anterior se define cada componente como:

**Modelo:** Representa la información del sistema que se opera, gestionando los accesos a dicha información, como consultas y actualizaciones, las peticiones de acceso o manipulación de información llegan al modelo a través del controlador

**Vista:** Presenta el modelo como un formato donde se puede interactuar, es decir son las interfaces de usuario o representación visual de los datos.

**Controlador:** Se encarga de comunicar e invocar peticiones del modelo cuando este hace alguna solicitud sobre la información, por ejemplo: editar, hacer un registro a la base de datos, enviar comandos de su vista, es decir el controlador es el intermedio entre la vista y el modelo.

## 8.2 BACKEND

**8.2.1 Spring Framework.** Es uno de los frameworks más populares basado en el lenguaje de programación Java, es de código abierto y reduce la implementación de código, facilita la seguridad, funcionalidad, se pueden establecer códigos de alto rendimiento, liviano o reutilizable, y así mismo se puede estandarizar, agilizar, resolver y emplear los problemas que surgen durante la programación.

Además, se implementó Spring MVC que es administrado también por Spring framework, que se enfoca en el diseño y desarrollo de aplicaciones que utilizan el modelo vista controlador, y está orientado a HTTP<sup>19</sup> y basado en servlets<sup>20</sup>. Los elementos más usados son las clases anotadas con `@Controller` o `@RestController` en el caso REST APIS.

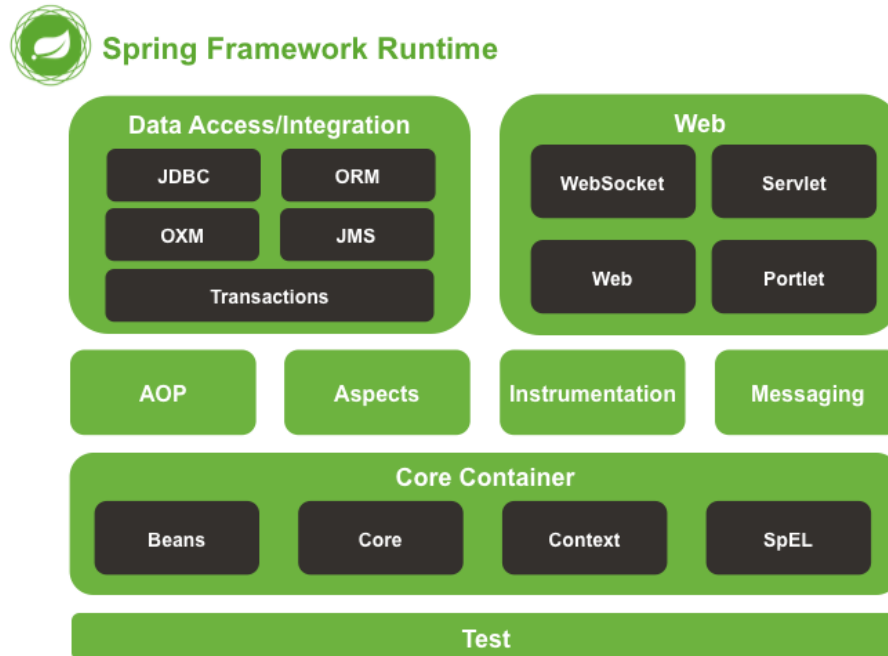
En la siguiente figura se presenta como está conformada la arquitectura Spring:

---

<sup>19</sup> Protocolo de transferencia de hipertexto: protocolo de comunicación que permite la transferencia de información en la Worl Wide Web.

<sup>20</sup> Módulos o clases escritas en Java que se utilizan en un servidor.

Figura 10. Arquitectura Spring



Fuente: Spring Framework, Disponible en: <<https://docs.spring.io/spring/docs/4.2.x/spring-framework-reference/html/overview.html>>

**8.2.2 Spring Boot.** Para lenguajes de programación existe una gran variedad, desde que se hizo su lanzamiento Java ha sido uno de los más utilizados, en este caso y para este proyecto en particular fue el lenguaje que se utilizó porque existían conocimientos previos de él, una de las principales razones, pero, además es un lenguaje orientado a objetos, es interpretado y compilado, es seguro entre otras características.

En las aplicaciones web existen diversas herramientas que facilitan el desarrollo, Spring boot es una herramienta que tiene como objetivo reducir al máximo la configuración inicial del desarrollador antes de empezar la codificación, Spring boot tiene la incorporación directa de servidores web como Apache Tomcat, lo que significó un gran beneficio, pues durante la decisión de que herramientas usar, saltarse la configuración del servidor suponía un alivio ya que se va directamente a la codificación y presionando run o play (dependiendo del IDE) tendríamos una

aplicación Spring en ejecución, además en Spring inicializ<sup>21</sup>, podemos elegir si queremos un proyecto maven o un proyecto Gradle, en que lenguaje (java, groovy o kotlin), la versión de spring boot y todas las dependencias o módulos que necesitamos y que trae integrados, tales como spring security, spring data, spring jpa, los driver de la base de datos, entre otras, y este nos descarga un proyecto ya configurado que solo debemos importar a nuestro IDE y empezar a trabajar sobre esa configuración base.

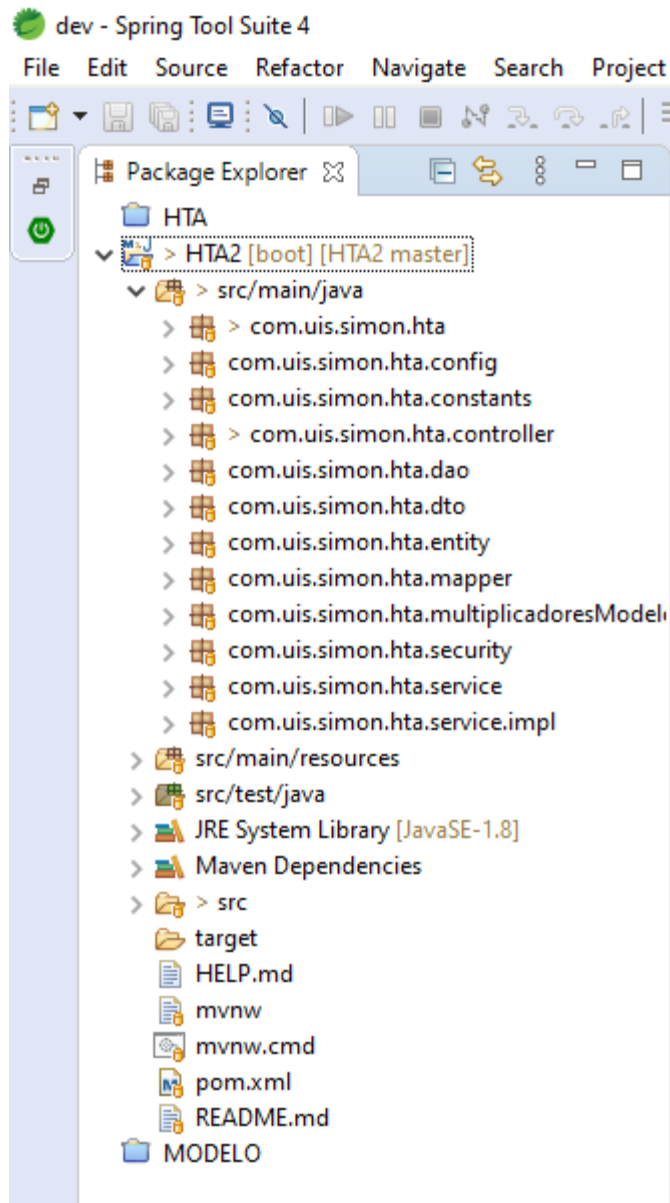
**8.2.3 Spring Tool Suite.** Es un IDE (Integrated Development Environment) herramienta de Spring que proporciona soporte sobre todo para aplicaciones basadas Spring Framework, en la página oficial se presentan tres opciones, ST for Eclipse, ST for Visual Studio Code y ST for Theia.

**8.2.4 Estructura del back.** Después de la fase de desarrollo el back el proyecto finalmente quedo constituido de la siguiente manera:

---

<sup>21</sup> <https://start.spring.io/>

Figura 11. Estructura del backend



Todos los paquetes tienen la forma `com.uis.simon.hta` que hace referencia al grupo de investigación, seguido por el nombre del paquete:

**Config:** Contiene la clase donde se configura la seguridad del proyecto

**Constant:** Contiene una clase donde están declaradas unas constantes que se usan para la seguridad del proyecto, como la generación de los tokens.

**Controller:** Es la capa donde se realizan las peticiones HTTP (REST) necesarias que son parte de la aplicación tales como:

POST: Crear

GET: Leer

PUT: Actualizar

DELETE: Eliminar

Hay un controlador por cada entidad, donde se definen las peticiones o métodos correspondientes a la misma.

**Dao:** Data Access Object, se encuentra las clases encargadas de realizar y gestionar las operaciones CRUD en la base de datos, también existe un Dao por cada entidad.

**Dto:** Data Transfer Object: Clases que se encargan de transferir los datos que se mapean, es decir crean un objeto con los atributos que pueden ser enviados o recuperados en una invocación, entonces permite crear clases que contienen atributos de múltiples entidades.

**Entity:** Contiene las clases donde se representan cada una de las tablas del modelo relacional, es decir son los objetos de persistencia, como: pacientes, visitantes, visitas entre otros.

**Mapper:** Contiene las clases que se usaron para mapear entidades a listas, y usarlas en DTO en caso de ser necesario.

**Multiplicadores:** Este paquete no es necesario dentro de la arquitectura del proyecto, pero tiene dos clases donde se definen multiplicadores que hacen parte del modelo en evolución.

**Security:** Contiene todas las clases relacionadas con la seguridad y generación de los token

**Service:** Contiene las interfaces donde se definen los métodos que serán implementados o que son necesarios para la lógica de la aplicación

**Service.Impl:** Tiene toda la lógica de la aplicación, la implementación de todos los métodos que se definen en los servicios incluyendo la clase donde se escribe el modelo.

Cabe aclarar que como en la primera versión se desarrolló un software de escritorio, lo que hicieron fue integrar el software de evolución donde se corre el modelo del riesgo de hipertensión dentro del mismo, para este caso como los módulos se desarrollaron en un entorno web, se implementaron las ecuaciones que conforman el modelo en una clase llamada `CorrerModeloServiceImpl`; primero se definió el método en una interfaz en el paquete `service`, y luego se puso en marcha toda la lógica del modelo dentro de la clase de implementación, ya que evolución no es compatible en entornos web.

## 8.3 FRONTEND

**8.3.1 Angular.** Es un framework para aplicaciones web desarrollado en typescript, de igual forma es de código abierto, se utiliza para crear y mantener una sola página (SPA - Single Page Application) es decir, la navegación y la carga de datos se hace dinámica y asíncrona con el servidor, es considerado muy favorable por su

creatividad y dinamismo; su objetivo es aumentar las aplicaciones en el navegador (Chrome, Firefox, opera, entre otros), fue creado y es mantenido por Google. Angular separa completamente el front-end y el back-end en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador) asegurando los desarrollos con rapidez, a la vez que posibilita modificaciones y actualizaciones.

Figura 12. Logo angular



Fuente: ABC TECH Group, Angular – faster and smaller Disponible en: <[https://miro.medium.com/max/1044/1\\*qRKMMAGVMLfp\\_q7RAux3Zg.png](https://miro.medium.com/max/1044/1*qRKMMAGVMLfp_q7RAux3Zg.png)>

En su primera versión se llamaba AngularJS, pasando el tiempo se estandarizo como Angular 2, Angular 3, Angular 4, Angular 5, Angular 6, Angular 7, para este proyecto se comenzó trabajando en la versión 8 y actualmente corre sobre la versión 9.0.2, así mismo ejecutándose al lado del cliente o servidor Node.js versión 13.9.0.

Figura 13. Angular cli

```
C:\Users\Asus>ng version

Angular CLI

Angular CLI: 9.0.2
Node: 13.9.0
OS: win32 x64

Angular:
...
Ivy Workspace:

Package                       Version
-----
@angular-devkit/architect     0.900.2
@angular-devkit/core          9.0.2
@angular-devkit/schematics    9.0.2
@schematics/angular           9.0.2
@schematics/update             0.900.2
rxjs                           6.5.3
```

**8.3.2 Node Js.** Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Al contrario que la mayoría del código JavaScript, no se ejecuta en un navegador, sino en el servidor. Node.js implementa algunas especificaciones de CommonJS.

**8.3.3 Características y componentes de angular.** Angular emplea el lenguaje TypeScript (creado por Microsoft), este tipo de programación es libre y de código abierto, incluyendo herramientas de programación orientado a objetos, y puede usarse para desarrollar aplicaciones JavaScript.

Las características más importantes de Angular son:

- Dividir el código por el uso de componentes y enrutamiento
- Genera código
- Código Universal
- Manejo de plantillas

- Herramienta de línea de comando llamada Angular CLI

En la vista se puede modificar el modelo, y en el modelo se puede modificar la vista, para la lógica de nuestros servicios y poder manejar la data en nuestra aplicación, se encapsulan nuestros componentes y servicios en los módulos o NgModules.

## Módulos

Un módulo o NgModule se declara para un conjunto de componentes y servicios para formar una unidad funcional, toda aplicación con angular tiene un *root module* o módulo raíz conocido como AppModule.ts, donde se tienen todos los mecanismos de arranque para que inicie nuestra aplicación.

## Componentes

Cada aplicación angular debe tener por lo menos un componente conocido, como *root component*, este conecta la jerarquía de componente con el Modelo de objetos del documento (DOM), define la clase que contiene datos y lógica de la aplicación, y está asociada a una plantilla HTML que define la vista que se va a obtener al final.

## Plantillas

Se combina HTML con el marcado de angular, realizando las modificaciones antes de la renderización la vista.

## Directivas Estructurales

Las directivas estructurales a elementos en el HTML que nos permiten añadir, manipular o eliminar elementos del DOM, son fáciles de reconocer ya que están anteceditas por un *asterisco* (\*) seguido del nombre de la directiva, por ejemplo:

`*ngFor = "let i = index" o *ngIf = "variable"`, entre otros.

## **Servicios**

Los servicios que tiene angular son un proveedor de datos, consumidos por los componentes en el proyecto, que delegan la responsabilidad de acceder a la información y realizar operaciones con los datos.

## **Rutas**

Las rutas nos proporcionan un servicio para navegar entre los estados de la aplicación, es decir procesa las rutas y determina cual vista es la que se debe mostrar en cada dirección asignada.

## **Angular material**

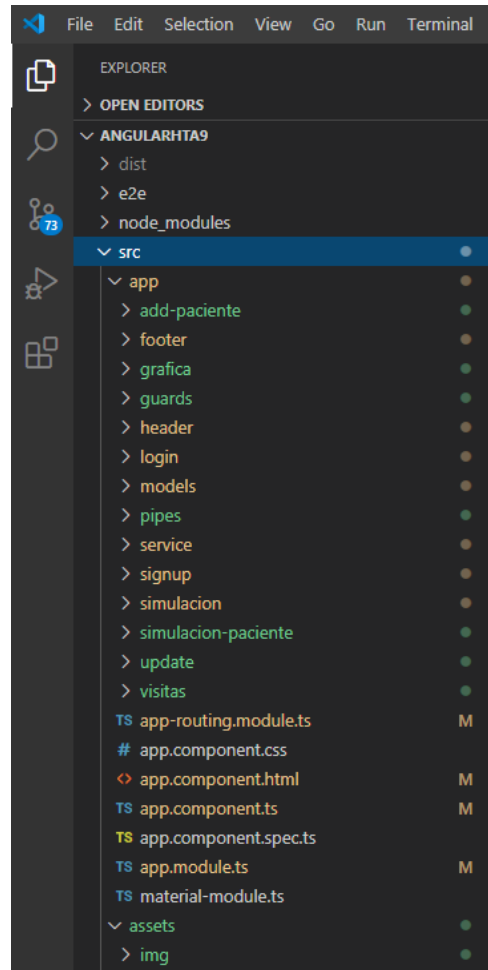
Es el componente de angular para darle estética a la vista, es similar a usar Bootstrap basado en la guía de diseño de material design con el fin de integrarse perfecto con angular, también fue creado por Google; angular material comprende de una gama de componente que implementa patrones de conexión de acuerdo con las especificaciones de material design, cuenta con varios controles tales como: Form Controls, Navigation, Layout, Buttons & Indicators, Popups & Modals y Data table.

Es fácil de incorporarlo a un proyecto angular, simplemente se añade un componente llamado material-module con todas las librerías disponibles de material e importándola al module, adicionalmente cuenta con temas ya preestablecidos que se adicionan al instalarlo construyendo dicha aplicación con el tema elegido, los temas son:

- Deep Purple & Amber
- Indigo & Pink
- Pink & Blue-grey
- Purple & Green

**8.3.4 Estructura del frontend.** A continuación, se muestra la estructura del front, cuyo código fue elaborado en el editor visual studio code.

Figura 14. Estructura del front



Angular cuenta con una estructura establecida para un buen manejo de esta, creado por primera vez este tiene un componente llamado **app** la cual tenía una vista simple con una demostración, esta **app** se mantiene en el proyecto para poder interactuar con la aplicación ya que es el módulo principal, en **app.modules** se anexan los componentes, las librerías o dependencias que se requieran, y además de esto para obtener los componentes de angular material se generó un archivo llamado **material-module.ts**, el cual se anexo e importo en **app.modules.ts**; luego de esto

debe existir la previa instalación de nodejs situada en una carpeta llamada **node\_modules**; en la carpeta **assets** tiene por defecto el anexo de imágenes o iconos representativos de la página diseñada, la carpeta **service** contiene el servicio el cual nos permite hacer peticiones HTTP, la carpeta **guards** se generó para proteger las rutas que no están autenticados.

Las otras carpetas son los componentes básicos para la construcción de la página web en Angular, es decir la parte visual en html (vista) y la parte funcional en typescript como el controlador.

#### 8.3.5 Resultado Final

Para conocer el resultado final del proyecto se puede dirigir al Anexo E donde se encuentra la presentación con la que se dio a conocer el proyecto a UISALUD, ahí podrá encontrar una breve explicación del proyecto y las imágenes de la interfaz además en la URL <http://simon.uis.edu.co/HTA/> podrá encontrar el prototipo en funcionamiento.

## 9. PRUEBAS

Uno de las cosas más importantes durante y al final del desarrollo es probar que el código funciona correctamente, existen muchos tipos de prueba que pueden ser aplicados en la ingeniería de software, pero de forma general tenemos dos tipos de pruebas: funcionales y no funcionales, dentro de las cuales se encuentran las pruebas unitarias, de aceptación, de carga, de estrés entre otras, para este prototipo particular no se considera que se requieran pruebas muy estrictas o de muchos tipos ya que este prototipo es sencillo y con operaciones CRUD, por lo tanto solo se implementaron unas pruebas unitarias a algunos métodos para probar que funcionan de manera correcta, y además se anexan las pruebas realizadas en Postman de cada una de las funcionalidades probadas con la base de datos, estas pruebas se encuentran en el Anexo F.

Asimismo, se aplicaron unas pruebas de usuario de forma remota con algunas personas que podrían ser usuarios del sistema y que estaban interesadas en conocer su estado de salud, debido al problema de salud mundial que existe actualmente las pruebas se hicieron de forma remota con los usuarios y se les pidió que llenaran una encuesta luego de conocer el aplicativo, estas pruebas se encuentran en el Anexo G.

## 10. INTEGRACIÓN

El último objetivo de este proyecto buscaba realizar una integración con el sistema de salud del paciente, en este caso UISALUD que es la entidad interesada, para esto existen muchas opciones a fin de integrar un sistema de salud, a continuación, algunas de ellas:

### 1. Transferencia de archivos

El sistema A crea un archivo (texto plano, binario etc) y lo pone en una ubicación donde el sistema B lo pueda leer. Es un estilo de integración muy usado.

Ventajas:

- Fácil de implementar (mientras el sistema de lenguaje o el framework sea capaz de generar el archivo)
- Estilo universal, cualquier sistema puede manipular archivos
- Abstracción entre sistemas, el sistema no se debe conocer uno de otro, mientras se pueda generar el archivo con un formato determinado.

Desventajas:

- Sistemas desincronizados, siempre va a ver un sistema más actualizado que otro, depende de cada cuanto se generen los archivos.
- Solo sirve para transferir datos si A desea ejecutar una funcionalidad de B, no podrá.

### 2. Base de datos Compartida

Es similar a la transferencia de archivos, la diferencia es que yo no escribo en un archivo si no en una base de datos.

Si A necesita pasar alguna información a B escribe en la base de datos en tablas predeterminadas y el sistema B en algún momento se conectará a la misma base de datos y va a leer la información.

Ventajas:

- Integración casi universal, si asumimos que estamos lidiando con bases de datos relacionales, muchos sistemas se pueden conectar a una base relacional.
- Es fácil de implementar
- Abstracción de sistemas
- Desacoplamiento
- Si B necesita A, A no tiene que estar corriendo, desde que se conecte a la base de datos

#### Desventajas:

- Problemas de rendimiento
- Si hay que modificar la base de datos, hay que ser muy cuidadosos y coordinar los cambios.
- Propiedad los datos, cuando las bases son accedidas por varios sistemas hay que aclarar la información, que información pertenece a que sistema porque se pueden modificar los datos y presentarse problemas de auditoría.
- Solo sirve para transferir datos

### 3. Invocación de procedimientos remotos

Un sistema A desea ejecutar algo en B, entonces hace un llamado a un procedimiento remoto y ese llamado genera una respuesta: ¿para que el llamado? ¿Quiero ejecutar una funcionalidad o solo pasar datos y cómo puedo hacerlo?

Para comunicarnos entre servicios o entre apps unas más modernas, otras más viejas podemos usar Rest, soap, web sockets o muchas tecnologías a la vez.

#### Ventajas:

- Encapsulación de los datos, cada sistema es dueño de su fuente de información no importa en que está hecha, los datos están encapsulados y

tienen sus dueños, asimismo si se quiere acceder se tiene que ir a través del sistema.

- Integramos con sistemas externos, Rest o Soop.
- Fácil de implementar porque las herramientas son muy robustas.

Desventajas:

- Alto acoplamiento entre los sistemas, ambos sistemas deben estar disponibles.

Este prototipo es considerado la base de un proyecto de investigación propuesto desde el grupo de investigación SIMON para UISALUD, razón por la cual se buscaba probar que la integración es posible y que existen varias opciones, para esto basados en las opciones anteriores (transferencia de archivos, base de datos, procedimientos remotos) se decidió usar la transferencia de archivos, UISALUD nos entrega información de pacientes y profesionales que realicen las visitas, los cuales en una reunión expresaban que debían ser un nutricionista o un fisioterapeuta, la información básica se almacena directamente en nuestra base de datos, y así, con el usuario y la contraseña que se les genera, pueden acceder al sistema y se les devuelve la información de las visitas que ellos van generando, se almacena en la base de datos y en un periodo de tiempo determinado se realiza la entrega de estos, como es un prototipo base la integración es muy sencilla, pero se dejan planteadas las demás opciones para el sistema que busca desarrollar la propuesta de investigación y con una integración más completa.

## 11. CONCLUSIONES

1. El producto de este proyecto constituyó un referente en la elaboración de la propuesta de investigación UISALUD-SIMON titulada *Prototipo funcional de telemonitoreo para promover y mantener la adherencia en pacientes de uisalud con diagnóstico de hipertensión arterial (hta) y diabetes mellitus 2 (dm2)*, financiada por UISALUD, y que, a la entrega de este informe está en trámite en la vicerrectoría de investigaciones.
2. Este prototipo al ser evaluado por los usuarios permitió observar su importancia para incentivar la reflexión de la entidad de salud sobre cómo llevar su proceso de atención primaria implementando visitas médicas domiciliarias. Además, en las pruebas de la herramienta, se apreció que los pacientes reflexionaban sobre la importancia que tienen los hábitos de vida y como los mismos impactan su salud y su riesgo de padecer de hipertensión arterial.
3. La implementación de este proyecto en un entorno web mejoró la eficiencia de la herramienta haciendo énfasis en el modelo que nos permite estimar el riesgo de la hipertensión arterial y la facilidad para operarla por parte de los usuarios. La nueva versión superó la limitación que tenía la primera versión para funcionar sobre diferentes sistemas operativos y brindar la posibilidad de hacer una integración con el sistema de información de UISALUD.
4. Este desarrollo permitió verificar que herramientas como Spring boot, angular 9.0 y Angular Material, facilitan la etapa de desarrollo de un proyecto permitiendo usar estructuras estándar para el diseño de la aplicación cumpliendo estándares básicos.

## 12. TRABAJO FUTURO

Para cualquier desarrollo futuro sobre todo se recomienda tener en cuenta la documentación que se anexa en este trabajo de grado. Durante el tiempo de desarrollo de este proyecto y teniendo en cuenta las reuniones que se llevaron a cabo con el personal de UISALUD cuya descripción y fechas específicas se encuentran en el Anexo H, tenemos las siguientes recomendaciones.

Con respecto al modelo:

- Se espera que el modelo sea validado con un conjunto de pacientes de afiliados a UISALUD para medir la relevancia clínica y el apoyo que este podría representar en la prevención de enfermedades como la hipertensión arterial, y trabajar de la mano con ellos para la implementación de modelos para otras enfermedades, como por ejemplo el estudio de Finnish para la diabetes y sobre todo que estos modelos se ajusten a la población sobre la cual será aplicados.
- Modificar el modelo del peso de manera que se le dé un enfoque distinto a la dieta, que no se base solo en la cantidad de calorías si no en el tipo de alimento del que proviene la caloría, ya sean proteínas, carbohidratos, azúcares u otros y el impacto que este tiene en nuestro peso, además que se tengan en cuenta las mediciones de sal, e impacto de estrés, la carga emocional y el rol de la familia.
- Considerar que el índice de masa corporal no nos indica que una persona se encuentre específicamente en un peso saludable o no saludable, para mejorar esto es necesario tener en cuenta el perímetro abdominal, que puede hacer más preciso el estado del peso de un paciente.

Con respecto a la plataforma:

- Desarrollar nuevos módulos teniendo que en cuenta que las visitas que se proponen las pueden realizar distintos tipos de profesionales: como un nutricionista, un fisioterapeuta, entre otros; al realizar la visita sería ideal tener un módulo donde se guarde o consulte la información pertinente al área del profesional y así iniciar el seguimiento a la salud del paciente.
- Añadir funcionalidades al sistema tales como una sección con noticias de salud, información relacionada con el tema, propuestas de dietas o actividades físicas que el paciente pueda descargar en base a su estado de salud actual.
- Mejorar aspectos de seguridad, relacionadas con el perfil del usuario (paciente o visitador), teniendo en cuenta que el prototipo está enmarcado en un proyecto de investigación que incluye integrar el sistema de información de UISALUD.

## BIBLIOGRAFÍA

ANDRADE SOSA, Hugo Hernando, *et al.* El modelado y la simulación en la escuela de preescolar a undécimo grado construyendo explicaciones científicas. Universidad Industrial de Santander, Bucaramanga, 2014.

ANDRADE SOSA, Hugo Hernando. *et al.* Pensamiento sistémico diversidad en búsqueda de unidad. Bucaramanga. Ediciones Universidad Industrial de Santander, 2007.

CARRILLO, Leidy M. & PORTILLA, Yeny P. Ambiente informático para la interacción médico-paciente en la intervención de factores de riesgo de hipertensión arterial Versión 1.0. Trabajo de grado, Universidad Industrial de Santander, Bucaramanga. 2017.

Framingham Heart Study, A project of the National Hear, Lung, and Blood Institute and Boston University, (2016), Hypertension [online]. Disponible en: <<https://framinghamheartstudy.org/fhs-risk-functions/hypertension/#>>

FUNDACIÓN NODEJS. Express Infraestructura web rápida, minimalista y flexible para Node.js [En línea] Express, 2019. Disponible en: <https://expressjs.com/es/>

GOOGLE. Angular Docs. Version 7.2.16-local+sha.51955bf. [En línea] Angular, 2019. Disponible en: <https://angular.io>

GOOGLE. Ng2-charts [En línea]. Disponible en: <<https://valor-software.com/ng2-charts/>>

GOOGLE. Spring Framework Documentation Version 5.2.7.RELEASE [En línea]: Disponible en: <<https://docs.spring.io/spring/docs/current/spring-framework-reference/>>

IEEE STANDARDS ASSOCIATION, IEEE STD 830-1998 Recommended Practice for Software Requirements Specifications, 2009.

LOPEZ RIVERA, Jesús Antonio. (2018). Programa para la detección, evaluación y tratamiento del paciente hipertenso de la región andina. [En línea] Disponible en: <<http://svmi.web.ve/wh/documentos/HTA-LASH.pdf>>

MINISTERIO DE SALUD DE COLOMBIA, Biblioteca Digital. [En línea] Disponible en: <<https://www.minsalud.gov.co/Paginas/default.aspx>>

RICHARDS, MARK. (2015) Software Architecture Patterns [En línea] Disponible en: <<https://www.oreilly.com/content/software-architecture-patterns/>>

STERMAN, John. Business Dynamics: System Thinking and Modeling for a Complex World. New York: McGraw-Hill Higher Education, 2000.