

SOFTWARE PARA DETERMINAR NIVEL DE TRIAGE HOSPITALARIO A TRAVÉS
DE INTELIGENCIA ARTIFICIAL.
ADVISOR IA

JEFFRY ENRIQUE LA ROSA PADRINO
OSCAR JAVIER MENDOZA RUIZ

UNIVERSIDAD ANTONIO NARIÑO
INGENIERÍA DE SISTEMAS
ESPECIALIZACIÓN INGENIERÍA DE SOFTWARE

BOGOTÁ
2020
JEFFRY ENRIQUE LA ROSA PADRINO
OSCAR JAVIER MENDOZA RUIZ

Proyecto de grado para optar al título de
ESPECIALISTA EN INGENIERIA DE SOFTWARE

DIANALIN NEME PRADA
MBA con énfasis en Gerencia de Proyectos

UNIVERSIDAD ANTONIO NARIÑO

INGENIERÍA DE SISTEMAS
ESPECIALIZACIÓN INGENIERÍA DE SOFTWARE
BOGOTÁ
2020

Nota de Aceptación

Presidente del Jurado

Jurado

Jurado

Bogotá 30-11-2020

*Gracias a Dios y a mi esposa Julieth
por acompañarme a vivir esta etapa
de la vida.*
(Jeffry La Rosa)

*Gracias a Dios y a mi Novia Mónica
por estar conmigo y apoyarme en
este camino.*
(Oscar Mendoza)

AGRADECIMIENTOS

Agradecemos a las personas implicadas en nuestra formación al decano de ingeniería de sistemas, junta directiva y docentes de la especialización de ingeniería de software, nuestros compañeros y familiares.

CONTENIDO

Pág.

Contenido

1. INTRODUCCIÓN	15
2. TITULO	16
3. FORMULACION Y DESCRIPCIÓN DEL PROBLEMA.....	17
3.1 Definición del problema	17
3.2 Justificación.....	17
4. OBJETIVO GENERAL	18
5. OBJETIVOS ESPECIFICOS.....	18
6. MARCO DE REFERENCIA	19
6.1 Estado del Arte	19
6.2 Impacto	20
6.3 Componente de Innovación.....	20
6.4 Marco teórico.....	21
7. METODOLOGÍA	27
8. PROCESO DEL SOFTWARE	29
8.1 Requerimientos funcionales	33
8.2 Requerimientos no funcionales	34
8.3 Diseño de arquitectura.....	35
8.3.1 Diagrama de despliegue	38
8.3.2 Casos de uso arquitecturalmente relevante	39
8.3.3 Diagrama de secuencia.....	40
8.3.4 Diagrama de clases.....	41
8.3.5 Arquitectura de alto nivel.....	42
8.4 Construcción.....	42
8.4.1 Creación de los repositorios de código fuente	43
8.4.2 Desarrollo de agente DialogFlow.....	43
8.4.3 Desarrollo de la base de datos.....	45
8.4.4 Desarrollo webhook	46
8.5 Pruebas	47
8.5.1 Pruebas de código estático	47

8.5.2 Pruebas Unitarias	48
8.5.3 Pruebas de carga.....	48
8.6 Instalación y configuración	49
8.6.1 Despliegue de aplicación.....	49
8.6.2 Despliegue de webhook	51
8.7 Aspectos de seguridad en la nube	52
9. ANEXOS	53
9.1 ANEXO A - Cronograma de actividades	53
9.2 ANEXO B- Servicios de seguridad AWS	54
10. CONCLUSIONES	55
11. REFERENCIAS-BIBLIOGRAFIA	56

LISTA DE TABLAS

	Pág
Tabla 1. Aplicaciones web de triage.	19
Tabla 2. Escala de color con descripción (Tomado de [18]).	21
Tabla 3. Cinco prioridades de triage (Tomado de [19]).	22
Tabla 4. Restricción #1 del sistema	30
Tabla 5. Restricción #2 del sistema	31
Tabla 6. Restricción #3 del sistema	31
Tabla 7. Framework utilizado para aplicación web	32
Tabla 8. Entornos web	33
Tabla 9. Especificación árbol utilidad	34
Tabla 10. Detalles de entornos web	36
Tabla 11. Método Librería SomosIoticos	47

LISTA DE GRÁFICAS

	Pág
Grafico1. Tiempos de atención en urgencias	29

LISTA DE FIGURAS

	Pág
Figura 1. Diagrama de decisiones de triage (Tomado de [19]).	23
Figura 2. Diagrama usuario final. (Autoría propia).	27
Figura 3. Árbol de utilidad resumido. (Autoría propia).	34
Figura 4. Componentes del sistema. (Autoría propia).	37
Figura 5. Diagrama de despliegue (Autoría propia).	38
Figura 6. Diagrama usuario final. (Autoría propia).	39
Figura 7. Diagrama de secuencia. (Autoría propia).	40
Figura 8. Diagrama de despliegue. (Autoría propia).	41
Figura 9. Arquitectura de alto nivel. (Autoría propia).	42
Figura 10. Repositorios proyecto Advisor. (Autoría propia).	43
Figura 11. Agente Dialogflow. (Autoría propia).	44
Figura 12. Integración Dialogflow Fullfilment. (Autoría propia).	45
Figura 13. Base de datos. (Autoría propia).	45
Figura 14. Estructura tabla centro hospitalario. (Autoría propia).	46
Figura 15. Repositorio de webhook. (Autoría propia).	46
Figura 16. Herramienta de análisis estático SonarCloud. (Autoría propia).	48
Figura 17. Entorno de prueba JMeter. (Autoría propia).	49
Figura 18. Carpeta workflow Github. (Autoría propia).	49

Figura 19. Procesos de despliegue Github Actions. (Autoría propia).	50
Figura 20. Aplicación Advisor. (Autoría propia).	51
Figura 21. Procesos de despliegue Webhook. (Autoría propia).	51
Figura 22. Entorno Elastic Beanstalk. (Autoría propia).	52
Figura 23. Cronograma de actividades del proyecto. (Autoría propia).	53
Figura 24. Servicios de seguridad AWS. (Autoría propia).	54

LISTA DE ANEXOS

	Pág
Anexo A. Cronograma de actividades	49
Anexo B. Servicios de seguridad AWS	50

GLOSARIO

FRAMEWORK: “En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta”. (Javier J. Gutiérrez., 2017)

GITHUB: “Es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git”. (Luciano Castillo, 2017)

WEBHOOK: “es un método de alteración del funcionamiento de una página o aplicación web, con callbacks personalizados”. (Wikipedia, 2020.)

RESUMEN

Este documento muestra la información referente a la implementación de la ingeniería de software en el diseño de un software que permite determinar el nivel de triage hospitalario con inteligencia artificial, dicho sistema fue desarrollado con las tecnologías php, javascript utilizando buenas prácticas y estándares. Se implementó el framework Angular y con tecnología de la empresa Google (Dialogflow) se desarrolló un chatbot que permite la comunicación con los usuarios, desplegándose en un entorno web administrado parcialmente por Amazon Web Services y manteniendo un repositorio de código fuente alojado en los servidores de la empresa Github, teniendo como finalidad la integración de diferentes tecnologías acorde al propósito del software, de manera de poder contribuir con los centros hospitalarios proporcionando una herramienta que disminuye los tiempos de diagnóstico de triage de manera significativa.

PALABRAS CLAVE: Dialogflow, Chatbot, Amazon Web Service, Triage, Inteligencia artificial, Software, Webhook.

1. INTRODUCCIÓN

El gran desafío de los sistemas de salud es la reducción en los tiempos de atención en las urgencias de los centros hospitalarios, esto fundamentado en el hecho de la preservación de la vida; A lo largo de la historia hemos podido ver como se han establecido diferentes mecanismos de atención y clasificación de los pacientes, teniendo como prioridad atender sin importar ningún tipo de condición social, económica, o religiosa entre otros, a los más afectados primeramente, es decir a quienes por su estado de salud se encuentren mas propensos a perder la vida.

El tiempo promedio para tener el primer contacto con un profesional del sistema de salud en los centros hospitalarios supera la barrera de los 30 min, a partir de ese momento es diagnosticado a través de un sistema donde se clasifican los pacientes según la patología que presente teniendo en cuenta que no todos los pacientes requieren atención inmediata; En este sentido el presente trabajo de grado se encuentra enfocado en poder brindar una propuesta de software, mediante el cual se pretende reducir los tiempos de espera para ser atendido durante esa primera etapa de evaluación en los centros hospitalarios. Darle a conocer a un paciente que presenta una sintomatología su estado no urgente de atención y recomendarle acudir a una cita externa permite que se puedan disminuir los tiempos de espera de los pacientes que realmente lo requieren.

Del tal manera que proveer un medio informático que pueda contribuir con la labor de agilizar la atención de los pacientes es una gran necesidad, que permita concurrencia de muchos pacientes, permaneciendo siempre disponible en internet para poder acceder a el mediante una interfaz intuitiva, considerando la facilidad al momento de hacer cambios a futuro y que integrarse a otros servicios no sea un proceso complicado; Es por ello que el presente trabajo describe cada etapa de ejecución de esta propuesta abordando el aspecto tecnológico, y el beneficio que esté ofrece a la sociedad mediante la contribución a la solución de problemas.

2. TITULO

Software para determinar nivel de triage hospitalario a través de inteligencia artificial.
Advisor IA.

3. FORMULACION Y DESCRIPCIÓN DEL PROBLEMA

3.1 Definición del problema

El gran número de personas que ingresan a urgencias en los centros hospitalarios contribuye en la mayoría de los casos a generar mayores tiempos de espera para ser atendidos teniendo como resultado que se compliquen, comprometiendo su nivel de salud y hasta la propia vida.

Enfocados en esta problemática, el proceso que se aborda con tecnología artificial, podríamos preguntarnos: ¿Es posible que, a los servicios de urgencias, solo asistan los pacientes que realmente necesitan de un médico urgente? Si lo logramos, habremos descongestionado estas áreas, optimizando así los tiempos de espera para los pacientes que lo requieren, y ayudando al uso racional y correcto de los recursos de este sector.

Teniendo en cuenta el contexto colombiano antes descrito y con el fin de apoyar, se diseña una solución informática y multimedia. Se plantea este trabajo en IA, con el fin de apoyar las instituciones prestadoras servicios de salud en Colombia, por medio de un proceso de clasificación de triage.

3.2 Justificación

Teniendo como medida los diferentes métodos de atención en la salud de Colombia, tener un medio electrónico que permita ayudar a los usuarios o pacientes a detectar de manera temprana enfermedades con patologías que se encuentren en triage 4 o 5 para así poder agilizar sus procesos de atención en las entidades de salud.

Triage IV: es una urgencia de nivel menor, que no compromete la vida del paciente ni pone en riesgo la pérdida de extremidades.

Triage V: es considerada como una no urgencia debido a que son problemas crónicos que presenta el paciente.

4. OBJETIVO GENERAL

Proporcionar un software que permita reducir los tiempos de atención en el diagnóstico de los niveles de triage, con el fin de disminuir el deterioro en la salud de los pacientes en los centros de urgencia.

5. OBJETIVOS ESPECIFICOS

1. Diseñar e implementar una arquitectura de software que permita la mantenibilidad y escalabilidad en el tiempo del proyecto ADVISOR IA.
2. Crear y diseñar un chatbot donde el usuario describe sus síntomas, el tiempo que lleva con ellos y diagnósticos previos, con el fin de direccionar al servicio de urgencias o a agendar una cita por consulta externa.
3. Realizar una página web con un asistente virtual donde el usuario consultará los datos y la información de las eps donde están afiliados, para luego validar las sedes de diferentes EPS, para orientarle cómo puede recibir los servicios en los puntos de atención.

6. MARCO DE REFERENCIA

6.1 Estado del Arte

En la actualidad existen miles de aplicaciones que hacen referencia a el estudio de los conceptos de los niveles triage, como determinarlos en sitio y dan una posible orientación de cómo manejar situaciones; Se realizo un análisis de los softwares más populares en la web, clasificándolos en por el tipo de función que ejecutan, encontrando que en su mayoría el enfoque principal es la formación a nivel de medicina en este tema del triage, esto se fundamentó en la siguiente tabla:

Tabla 1

Aplicaciones web de triage

Web app	Creador	Tipo	Ubicación
Triage Hospitalario	Superaplicacionex	Formación Medicina	Google Playstore
Health Interactive	Loma Linda University	Educativo/Tipo Juego	Google Playstore
Triage	TimeCapp	Toma de historial	Google Playstore
Triage Master	Amateur night programming	Formación Medicina	Google Playstore
STD Triage	Idoc24 inc	Análisis dermatológico	Google Playstore
Join Triage	Allm inc	Atención Medica	Google Playstore
EmisHealth	EMIS	Atención Medica/Distancia	https://www.emishealth.com/
Triage	Desconocido	Atención Medica/Distancia	https://www.triage.com/

Nota. Esta tabla muestra las aplicaciones actuales de triage.

En tal sentido, y bajo un análisis unitario de cada aplicación se pudo observar que son pocas las aplicaciones que ofrecen atención inmediata en la determinación de los niveles de triage; teniendo en cuenta que en mucho de los casos se pide que adjunten evidencias de la situación y puede demorar hasta 2-3 días en dar respuestas de las condiciones patológicas y estado de salud de los pacientes.

6.2 Impacto

En Bogotá la atención en el servicio de urgencias de hospitales presenta fallas en el proceso de valoración, siendo el tiempo de espera y la sobreocupación los factores que más influyen en la percepción de calidad de éste servicio médico; es por ello que a través del desarrollo de ADVISOR IA se descongestionaría en gran manera, permitiendo la disminución en los tiempos de espera en las salas de urgencias, educado a la vez a la población de que no todas las patologías requieren atención inmediata, sino que pueden ser tratadas bajo consultas externas programadas.

Este software brindaría una disponibilidad de 24 horas, por lo que cualquier persona que presente sintomatología va a tener el alcance de un primer diagnóstico, bajo un interfaz web que no requiere autenticación y brinda información de los centros asistenciales a los que puede acudir sin necesidad de pagar.

6.3 Componente de Innovación

La inteligencia artificial es el factor clave de este proyecto y, es lo que marca la diferencia respecto a todas las aplicaciones que ya existen en el mercado, pues en la mayoría de estos casos se necesita una respuesta inmediata, la cual muchos sistemas de aplicaciones no la ofrecen. Un chatbot implementado al sistema de salud, en el área de urgencias, permitirá clasificar por los síntomas indicados por los pacientes, el nivel de triage en el que requiere ser atendido, en tiempo real, lo cual se puede hacer desde cualquier dispositivo con conexión a internet, descongestionando así de forma sustancial las salas de emergencia, entendiendo que en la mayoría de los casos, los ciudadanos sobre-estiman su valoración y abordan sistemas de urgencias sin necesidad, pudiendo hacer citas programadas.

Las nuevas tecnologías van enfocadas a la optimización de procesos. La forma como se están tomando las decisiones a través de programas de conducta aprendida, evolucionan totalmente cualquier modelo de negocio, y el aprovechamiento de éste. Y lo más importante, permitiría salvar vidas al reducir los tiempos de atención y priorización de los casos.

La implementación de las nuevas tecnologías de procesamiento de lenguaje natural (NPL) como es el servicio brindado por la empresa Google a través de su componente Dialogflow entrega valor, a la cadena de procesos establecidos dentro del software.

6.4 Marco teórico

La empresa desarrolladora de Software YeePLY realizó un estudio donde cita: “Existen más de 75.000 apps de salud, aunque más de la mitad de éstas no llegan ni a las 500 descargas. Aun así, el desarrollo de aplicaciones de salud es un área que aún tiene mucho que ofrecer y que todavía no se ha explorado ni explotado a fondo.” (YEEPLY, 2014).

La empresa Inneva Pharma (2017) desarrolló exitosamente una aplicación móvil para mejorar el servicio de urgencias en Madrid, España: “El proyecto de app Urgencia Responsable de la empresa Inneva Pharma, es una herramienta educativa de orientación para que los pacientes sepan dónde pueden acudir en función de su estado y evitar el colapso de los servicios de urgencias en los hospitales. También permitirá evaluar la pertinencia de un tipo de atención concreta dependiendo de los datos del paciente, los síntomas generales y los recursos existentes, con la posibilidad de integrar la información de su historia clínica electrónica.” (ESALUD, 2017)

Triage de Urgencias

Sistema de selección y clasificación de pacientes, que centra su prioridad en discernir entre la necesidad del paciente que acude al centro de atención y los recursos que demandará el mismo. Es un proceso dinámico, que cambia tan rápidamente como lo puede hacer el estado del paciente, su evolución puede ser lenta y no necesariamente mortal. Puede ser observado en la Tabla 4 la escala de color aplicada a este tipo de clasificación triage.

Tabla 2

Escala de color con descripción

Clasificación	Color	Descripción	Tiempo Máximo de espera (min)	¿Qué hacer?	Tipos de Consulta
I	Rojo	Atención inmediata	0	Apoyo	Urgencia
II	Naranja	Muy urgente	10		
III	Amarillo	Urgente	60	Espere	Prioritaria
IV	Verde	Normal	120		
V	Azul	No urgente	240	Consulte	Consulta General

Nota. Identificamos la escala de color (Información de triage tomada de [18])

En la ley colombiana se establecen cinco prioridades de triage (ver Tabla 1). Sin embargo, a pesar de que el proceso de valoración al igual que los niveles de clasificación triage son invariantes, los tiempos de espera de los pacientes sí pueden variar debido a cambios de estado clínico o la llegada de nuevos pacientes con nivel de prioridad superior. El priorizar

la atención del paciente, garantiza que el tratamiento sea acorde a las necesidades clínicas con las cuales se asiste al centro especializado prestador de servicio.

Tabla 3
Cinco prioridades de triage

Escala	Tiempo de respuesta	Descripción
Prioridad I Reanimación	Atención Médica y de enfermería: inmediato, simultánea a la valoración.	Paciente que presenta condición que amenaza la vida; el paciente requiere una intervención médica inmediata. Se incluyen en esta categoría pacientes con dificultad respiratoria severa, estado de inconciencia o ausencia de signos vitales, debido a trauma mayor, problemas cardiorrespiratorios o neurológicos.
Prioridad II Emergencia	Atención de enfermería inmediato. Atención médica: 15 minutos.	Paciente con estabilidad ventilatoria, hemodinámica y neurológica, cuyo problema representa un riesgo potencial de amenaza a la vida o pérdida de una extremidad u órgano si no recibe una intervención médica rápida; por ejemplo pacientes en estado de agitación, dolor torácico, dolor abdominal, síntomas asociados con diabetes descompensada, algunas cefaleas, trauma o fiebre alta en niños, otras dolencias como vómito y diarrea, dolor cólico renal, amputación traumática. También se incluye todo tipo de dolor severo (nivel 7-10) y afecciones en las cuales el tiempo es crítico para iniciar el tratamiento.
Prioridad III Urgencia	Atención médica y de enfermería: menor de 30 minutos.	Paciente con estabilidad ventilatoria, hemodinámica y neurológica con condiciones que pueden progresar a problemas serios que requieren intervención de emergencia. Regularmente se asocian con molestias relevantes que interfieren en el trabajo o las actividades de la vida diaria. Son ejemplos de estos síntomas cefalea, dolor torácico, asma leve a moderada, sangrado leve a moderado y síntomas asociados con diálisis. También se incluye todo tipo de dolor moderado y afecciones en las cuales el tiempo es crítico para iniciar el tratamiento.
Prioridad IV Urgencia Menor	Atención médica y de enfermería: menor de 60 minutos.	Condiciones relacionadas con la edad del paciente, angustia o deterioro potencial o complicaciones, que se beneficiará de la intervención o de tranquilizarlo dentro de 1-2 horas. Incluye síntomas como dolor torácico (no sugestivo de síndrome coronario agudo), dolor de cabeza, dolor abdominal y depresión.
Prioridad V No Urgente	Tiempo de atención: 120 minutos.	Condiciones que pueden ser agudas pero no comprometen el estado general del paciente y no representan un riesgo evidente; también condiciones que hacen parte de problemas crónicos sin evidencia de deterioro. Por ejemplo, trauma menor estrés emocional e inflamación de la garganta. La atención puede ser postergada y el paciente puede ser referido a consultorio anexo.

Nota. Esta tabla muestra la descripción de los niveles de triage (Tomado de [19]).

El ministerio de salud colombiano determinó [19], el proceso para llevar a cabo la clasificación triage que debe ser implementado en cada una de las EPS.

Proceso de triage

El proceso de triage, es decir la clasificación de pacientes usando un árbol de decisiones, es llevado a cabo generalmente por una enfermera o enfermero. Estos son los encargados de realizar las siguientes actividades relacionadas con el triage: Valora y clasifica el paciente según su condición clínica. Recibe el paciente gravemente enfermo o lesionado en el sitio de llegada, realizando una adecuada valoración e inicia el manejo. El paciente ingresa, en caso de ser prioridad I ó II se atienden inmediatamente, para los casos III, IV y V, aguardan en la sala de espera hasta 2 horas aproximadamente, dependiendo de la disponibilidad de recurso humano para ser atendido.

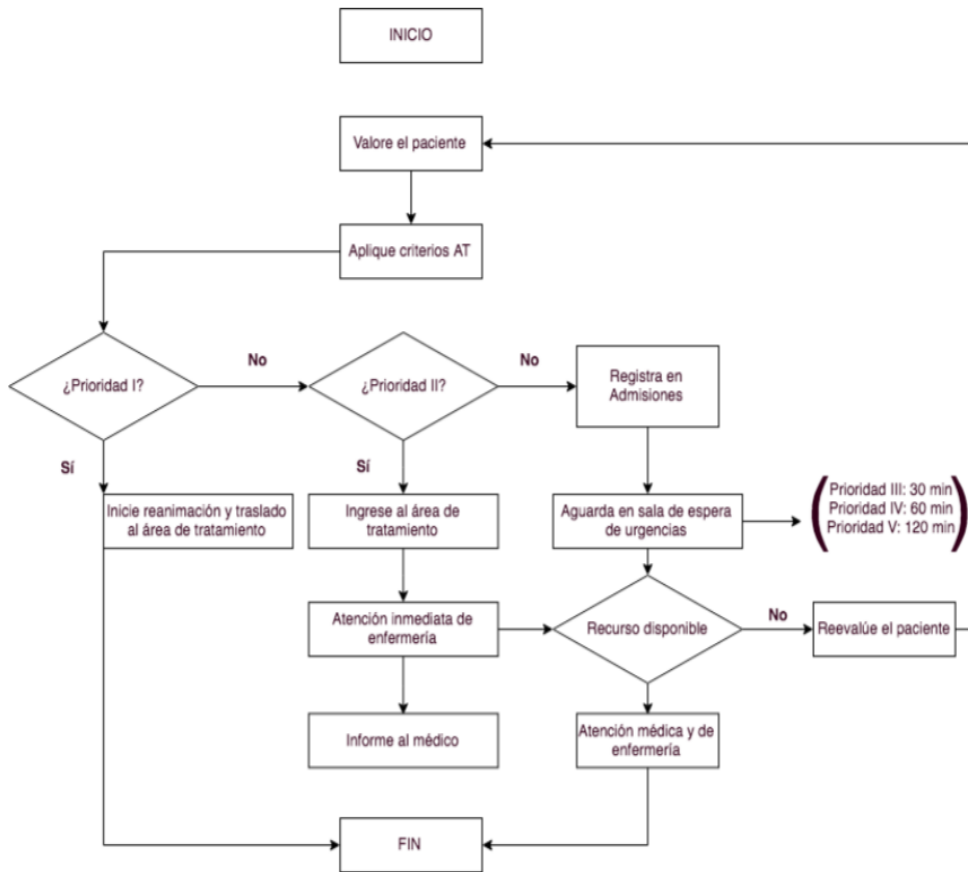


Figura 1. Diagrama de decisiones de triage (Gráfico tomado de [19])

Objetivos del triage

- Asegurar una valoración rápida y ordenada de todos los pacientes que llegan a los servicios de urgencias, identificando aquellos que requieren atención inmediata.
- Clasificar a los pacientes de acuerdo a su prioridad clínica y el personal disponibles en el ente hospitalario.
- Disminuir el riesgo de muerte, complicaciones o discapacidad de los pacientes que acuden a los servicios de urgencia.
- Informa tanto al paciente como a sus familiares de la condición de salud en la cual se encuentra y el estado de disponibilidad de atención en el momento para disminuir la ansiedad de ellos.

Software

“Según la Real Academia Española, el software es un conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora.” Perez, Julian., Gardey, Ana.(Julio 2020).Definicion.de: Definición de software (<https://definicion.de/software/>)

Ingeniería de software

“La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de programas informáticos, más conocidos como softwares”. SystemsGroup.(Agosto 2020).La ingeniería de software. (<https://systemsgroup.es/tecnologias-de-la-informacion/la-ingenieria-de-software-que-es-y-que-utilidad-tiene/32363/>)

Chatbot

Tecnología que permite que un usuario pueda mantener una conversación con un programa informático. Lo incluiremos dentro de este proyecto, para que el usuario final reciba respuestas precisas y ajustadas a su caso específico, y así mismo pueda ser direccionado con mayor certeza, a los servicios que realmente necesita recibir.

Inteligencia artificial

Programa de computación diseñado para realizar determinadas operaciones que se consideran propias de la inteligencia humana, como el autoaprendizaje.

Dentro del proceso de desarrollo del software se utilizará la herramienta DialogFlow, la cual permitirá el flujo de la conversación con el chatbot.

Programación Orientada a objetos (POO)

Técnica que aumenta considerablemente la velocidad de desarrollo de los programas gracias a la reutilización de los objetos. El objeto es el conjunto complejo de datos y programas que poseen estructura y forman parte de una organización.

Conceptos de Programación Orientada a Objetos

- *Clase*: Representa características, atributos y métodos de un conjunto de objetos del mismo tipo.
- *Atributos*: son las propiedades que describen a los objetos pueden ser datos.
- *Métodos*: Son las acciones, operaciones, procedimientos y funciones los cuales tenemos que aplicar a los atributos de los objetos que ocupamos.
- *Objeto*: Como ya sabemos un objeto es la instancia de una clase donde los objetos pueden contener atributos y métodos que manejan la solución al problema que queremos resolver.

Aspectos de la programación orientada a objetos

Abstracción: Representa la esencia de los objetos, capturando sus comportamientos y sirviendo de modelo que puede modificar su estado, a su vez permite comunicarse con otros objetos sin revelar como se es el proceso interno de implementación de las características que le subyacen. El proceso de abstracción se caracteriza por determinar los comportamientos que son comunes para definir las entidades que representan en el mundo real.

Encapsulamiento: Consiste en asociar todos aquellos elementos que semejan mayor relación a la entidad, considerando la misma abstracción.

Modularidad: Es una propiedad que permite aplicar una subdivisión, las cuales se denominan módulos, intentando que cada una sea lo más independiente posible de la aplicación.

Principio de ocultación: Es la forma de como un objeto puede presentarse a los demás por medio de una interfaz ocultando sus propiedades internas; esto permite que otros objetos no puedan conocer dichas propiedades y modifique algún estado de manera no deseable. Es decir, a través del ocultamiento aseguramos que solo los métodos internos tienen acceso a dichas propiedades.

Polimorfismo: Esta propiedad hace referencia a que los objetos puede adoptar comportamientos diferentes dependiendo de su llamado, entendiéndose que dichos comportamientos pueden tener el mismo nombre, pero estar asociados a objetos distintos.

Herencia: Permite crear objetos especializados a partir de objetos preexistentes, heredando las propiedades y comportamientos de todas las clases a las que pertenecen.

Diagramas UML

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Recordemos que un modelo es una representación simplificada de la realidad; el modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema. Los diagramas que corresponden al modelo de UML más usados son los siguientes:

- Diagrama de Clase

Los diagramas de clases describen la estructura estática de un sistema. Las cosas que existen y que nos rodean se agrupan naturalmente en categorías. Una clase es una categoría o grupo de cosas que tienen atributos (propiedades) y acciones similares.

- Diagrama de casos de uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario.

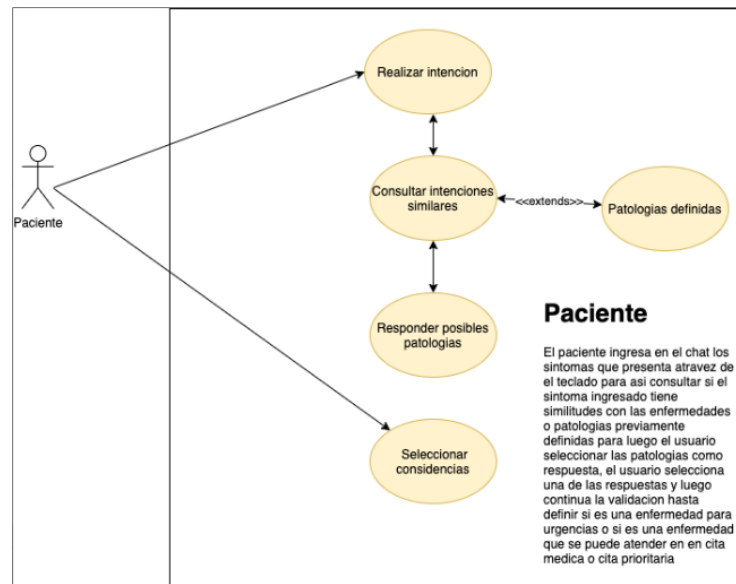


Figura 2. Diagrama usuario final. (Autoría propia).

- Diagrama de secuencia

Este diagrama permite modelar de manera gráfica la interacción que existe entre los objetos de un sistema.

- Diagrama de componentes

Permite representar un sistema denominado software en partes comúnmente llamadas componentes los cuales representan las estructuras de alto nivel y pueden ser de tipo software y hardware a la vez.

- Diagrama de despliegue

El Diagrama de Despliegue es un tipo de diagrama de UML que se utiliza para modelar la disposición física de los artefactos software en nodos (usualmente plataforma de hardware).

7. METODOLOGÍA

Para el desarrollo de este proyecto se utilizó la metodología ágil XP (Programación extrema), la cual se basa en la simplicidad, la comunicación y la realimentación del código desarrollado.

La metodología XP se implementó en sus cuatro etapas principales:

Análisis

Extraer los requisitos de un producto de software es la primera etapa para crearlo. A continuación, se listan las actividades que se realizaron para la etapa de análisis:

- Especificar requerimientos:
 - Levantar requerimientos.
 - Describir requerimientos.
 - Priorizar requerimientos.
 - Realización de encuesta.

- Determinar alcance:
 - Seleccionar los aspectos claves que debe incluir la aplicación.
 - Validar alcance.

Diseño

Se refiere a determinar cómo funcionará de forma general sin entrar en detalles. Consiste en incorporar consideraciones de la implementación tecnológica, como el hardware, la red, etc. Se definen los Casos de Uso para cubrir las funciones que realizará el sistema, y se transforman las entidades definidas en el análisis de requisitos en clases de diseño, obteniendo un modelo cercano a la programación orientada a objetos.

Desarrollo

Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no es necesariamente la porción más larga. La complejidad y la duración de esta etapa está íntimamente ligada al o a los lenguajes de programación utilizados.

A continuación, se listan las actividades que se llevarán en la etapa de desarrollo:

- Implementar requerimientos funcionales.

- Verificar cumplimiento de requerimientos no funcionales.

Pruebas

Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación. Una técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo.

El proceso será iterativo para las cuatro fases en lapsos cortos de 2 semanas por entrega con reuniones diarias y programación en par, permitiendo así el avance del proyecto minimizando los riesgos.

8. PROCESO DEL SOFTWARE

El proceso de desarrollo del software se encuentra fundamentado en la metodología implementada, partiendo en una fase de análisis donde se contemplaron los tiempos de atención por las categorías mostradas en la siguiente grafica.

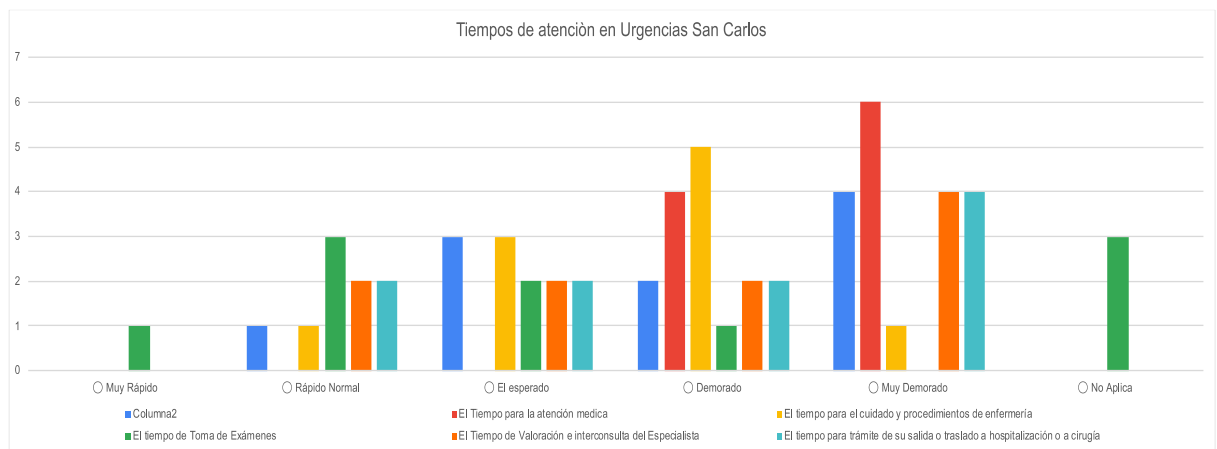


Gráfico 1. Tiempos de atención en urgencias (Autoría propia).

Teniendo en cuenta que el proyecto está delimitado a responder a de acuerdo a las patologías definidas a continuación por un profesional de la salud el cual se encuentra con el conocimiento intelectual y profesional para definir las:

Rinitis Alérgica Triage 5: Es una reacción alérgica que provoca ojos lloros y comezón y estornudos.

Conjuntivitis Triage 5: Inflamación o infección de la membrana externa del globo ocular y el párpado interno.

Tinitis Triage 5: Es comúnmente un sonido o timbre en el oído, se conoce también como acufeno.

Cefalea Cronica Triage 4: Es un grupo de cefaleas (Dolor de cabeza) característica por la frecuencia casi diaria presentando dolores más de 15 días al mes.

Lumbalgia Cronica Triage 4: Es un dolor que se presenta en la espalda baja de la espalda cuyo origen es con la estructura musculo esquelética de la espalda.

Odontalgia Triage 4: Dolor o inflamación en los dientes o alrededor de ellos producido por caries dental o infección.

Adicionalmente se establecieron criterios de restricciones a cumplirse en esta etapa del proyecto, considerando las limitaciones de elaboración del trabajo:

Tabla 4

Restricción #1

ID Restricción: R-1	Nombre: Uso Arquitectura de en capas	Tipo: Tecnología (X) Negocio ()
Descripción:	Se debe usar una arquitectura de capas.	
Establecida por:	Gerencia Advisor IA.	
Alternativas:	Como opción se puede cambiar a n-tier pero se requiere que el procesamiento de lenguaje natural será propio o bajo una tecnología open source bajo integración.	
Observaciones:	Ninguna	

Nota. Restricción #1 del sistema. (Autoría propia).

Tabla 5
Restricción #2

ID Restricción: R-2	Nombre: Uso de plataforma Google Dialogflow.	Tipo: Tecnología (X) Negocio ()
Descripción:	Se debe utilizar la plataforma de la empresa Google, Dialogflow.	
Establecida por:	Gerencia Advisor IA.	
Alternativas:	Rasa, Amazon lex.	
Observaciones:	Se puede hacer una migración de Dialogflow a Rasa mediante adaptaciones al json exportado. Adicionalmente para la ejecución de este proyecto la capa gratuita de Dialogflow permite solo 180 solicitudes por minuto.	

Nota. Esta tabla muestra la segunda restricción referente a Google. (Autoría propia).

Tabla 6
Restricción #3

ID Restricción: R-3	Nombre: Tiempo y personal requerido para la implementación del software.	Tipo: Tecnología () Negocio (x)
Descripción:	El tiempo de desarrollo del software deberá ser máximo de 6 meses con un equipo compuesto de 2 personas.	
Establecida por:	Gerencia Advisor IA.	
Alternativas:	Limitar el alcance del proyecto.	
Observaciones:	Debido al tiempo establecido por la universidad para cumplir los plazos de la especialización fue tomada esta restricción.	

Nota. Esta tabla muestra la restricción asociada a los recursos. (Autoría propia).


Dentro del desarrollo del software se utilizaron los siguientes frameworks y entornos:

- Frameworks

En el proyecto se decidió la utilización de Angular como framework de trabajo por su amplia gama de componentes que permiten ajustarse a todos los procesos asociados a la aplicación.

Tabla 7

Framework utilizado para aplicación web

FRAMEWORKS	DESCRIPCION
<p data-bbox="488 806 597 835">Angular</p> 	<p data-bbox="797 879 1458 1018">Es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página.</p>




Nota. Esta tabla muestra el Framework utilizado en la aplicación web. (Autoría propia).

- Entornos

Entendiendo que el fundamento web de esta aplicación contempla se ha decidido realizar la implementación de la aplicación tomando como servicios en la nube empresas reconocidas en el sector, teniendo la posibilidad del aprovechamiento de los prolongados periodos de prueba que en la actualidad ofrecen, permitiendo así la utilización de todos sus servicios.

Estos entornos permiten la integración de manera fluida entre ellos lo cual facilita su entendimiento e integración con mínimos esfuerzos y resaltando la manera como abordan los temas de seguridad entre sí; garantizando protocolos que minimizan los posibles ataques y pérdida de datos.

Tabla 8
Entornos web

ENTORNOS	DESCRIPCION
<p>Amazon Web Service</p> 	<p>Es una colección de servicios de computación en la nube pública que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por Amazon.com.</p>
<p>Google Dialogflow</p> 	<p>Es una plataforma de comprensión del lenguaje natural utilizada para diseñar e integrar una interfaz de usuario conversacional en aplicaciones móviles, aplicaciones web, dispositivos, bots, sistemas interactivos de respuesta de voz, etc</p>
<p>GitHub</p> 	<p>Es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git.</p>

Nota. Esta tabla muestra los 3 entornos web utilizados. (Autoría propia).

8.1 Requerimientos funcionales

El software enfocado en un chatbot con inteligencia artificial deberá cumplir con la función de emitir un diagnóstico del nivel de triage; Teniendo en cuenta que debe cumplir con algunas características relevantes como son las siguientes:

- El chat debe ser capaz de interactuar con la persona, dando una respuesta en tiempo real.
- El sistema debe poder escalar ante la concurrencia de usuarios.
- Debe contar con una alta disponibilidad.

- El sistema debe poder brindar al usuario facilidad de al momento de interactuar con el usuario, siendo implementado en una sola página y siendo el chatbot quien guie la conversación.

Es importante resaltar que la aplicación es netamente informativa, por lo cual no requiriere el almacenamiento de datos personales o clínicos, teniendo en cuenta que los aspectos legales que estos implican, por ende, el sistema informara al paciente que los datos no serán guardados en sistema.

8.2 Requerimientos no funcionales

En esta sección se establecieron los requerimientos no funcionales en función a los atributos de calidad con base a los cuales ha de diseñarse la aplicación Advisor IA, tomando en cuenta las necesidades de los stakeholders.

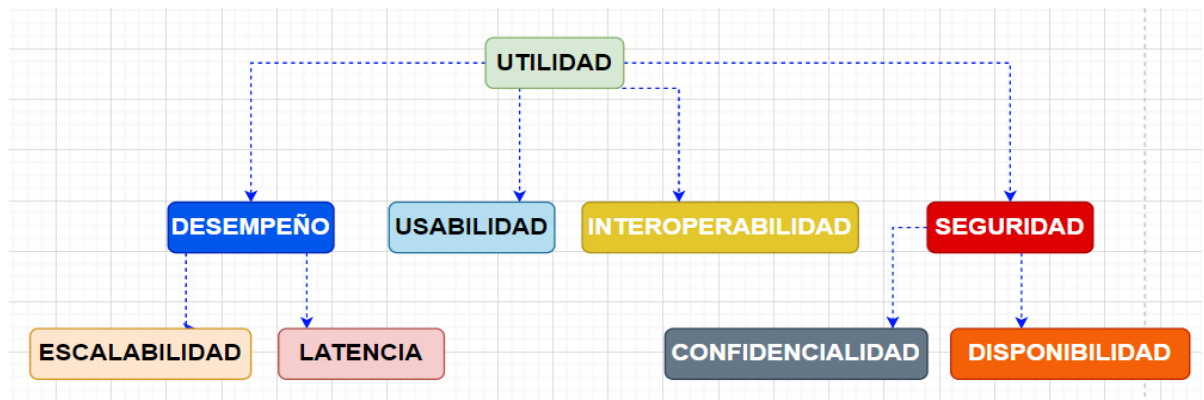


Figura 3. Árbol de utilidad resumido. (Autoría propia).

Tabla 9

Especificación árbol utilidad

Atributo de Calidad:		Seguridad
	ID	Descripción
Autenticación	S001	La aplicación no requiere de nivel de autenticación a nivel de usuarios porque es de tipo informativa.

Atributo de Calidad:		Disponibilidad
	ID	Descripción
	D002	El sistema central debe estar disponible el 99.99% de tiempo al año (52,6 minutos fuera).
Atributo de Calidad:		Interoperabilidad
	ID	Descripción
	I001	El sistema debe poder consultarse en los navegadores Chrome 84.0.4147.125, Mozilla Firefox 79.
	I002	El sistema debe poder consultarse en teléfonos móviles con soporte HTML.
Atributo de Calidad:		Usabilidad
	ID	Descripción
	U001	El sistema debe consultarse de manera ágil e intuitiva. Debe contar con el chatbot en la página principal.
Atributo de Calidad:		Desempeño
Escalabilidad	ID	Descripción
	D001	El sistema debe poder escalar para atender 100 transacciones concurrente de 20 segundos cada una.
Latencia	ID	Descripción
	D002	El chatbot debe de emitir una respuesta en menos de 20 segundos.

Nota. Esta tabla especifica los atributos de calidad. (Autoría propia).

8.3 Diseño de arquitectura

Esta fase tuvo como objetivo principal diseñar una arquitectura que se ajustara a las necesidades de los diferentes usuarios, de acuerdo a los diferentes casos de usos establecidos previamente.

En la presente sección del documento se establecieron los modelos, diagramas y artefactos de diseño que son relevantes para la construcción del software. La realización de estos

diagramas representó para el grupo de trabajo una abstracción sobre los componentes que representan los aspectos físicos y lógicos del sistema. Cabe resaltar que los diagramas aquí mostrados demuestran el comportamiento ideal, la manipulación y el almacenamiento de la información y la interacción con el usuario.

La arquitectura web fue diseñada bajo entorno web bajo entornos web que en la actualidad presentan planes gratuitos de prueba por un tiempo limitado dentro de (1) año, por lo que se considera de carácter académico y representativo. Adicionalmente se acondicionó el entorno local de desarrollo que cuenta con versionado de código fuente.

Tabla 10

Detalles de entornos web

ENTORNO WEB	AWS	DIALOGFLOW	GITHUB	SONARCLOUD
Propósito de uso	Alojamiento y despliegue continuo de la aplicación mediante servicios web ofrecidos en la capa gratuita	Integración de la interfaz gráfica y utilización del procesamiento de lenguaje natural bajo la tecnología desarrollada por Google.	Almacenamiento y despliegue de repositorios de código fuente.	Análisis de código estático
Características	Host Dominio Pipelines BD Versiones	Manejo de intenciones. Manejo de contextos Integraciones con terceros	Repositorio públicos y privados 3000 min de despliegue con Github Actions Integración con herramientas de análisis de código estático	Métricas de análisis de código estático. Sugerencias de mejora de código Integraciones
Versión	FreeTier	2.0	Git Hub, Inc.	Gratuita

Nota. Esta tabla especifica las características de los entornos web. (Autoría propia).

Los componentes principales del sistema distribuidos por servidores en los cuales se encuentran alojados; En AWS server se encuentran el componente de página web (alojada en un bucket s3), el componente webhook (alojado en Elastic Beanstalk), y el componente de base de datos (alojado en RDS); Por otra parte, tenemos al componente Dialogflow ubicado en el servidor de Google, el cual tiene comunicación directa con la página web y el webhook.

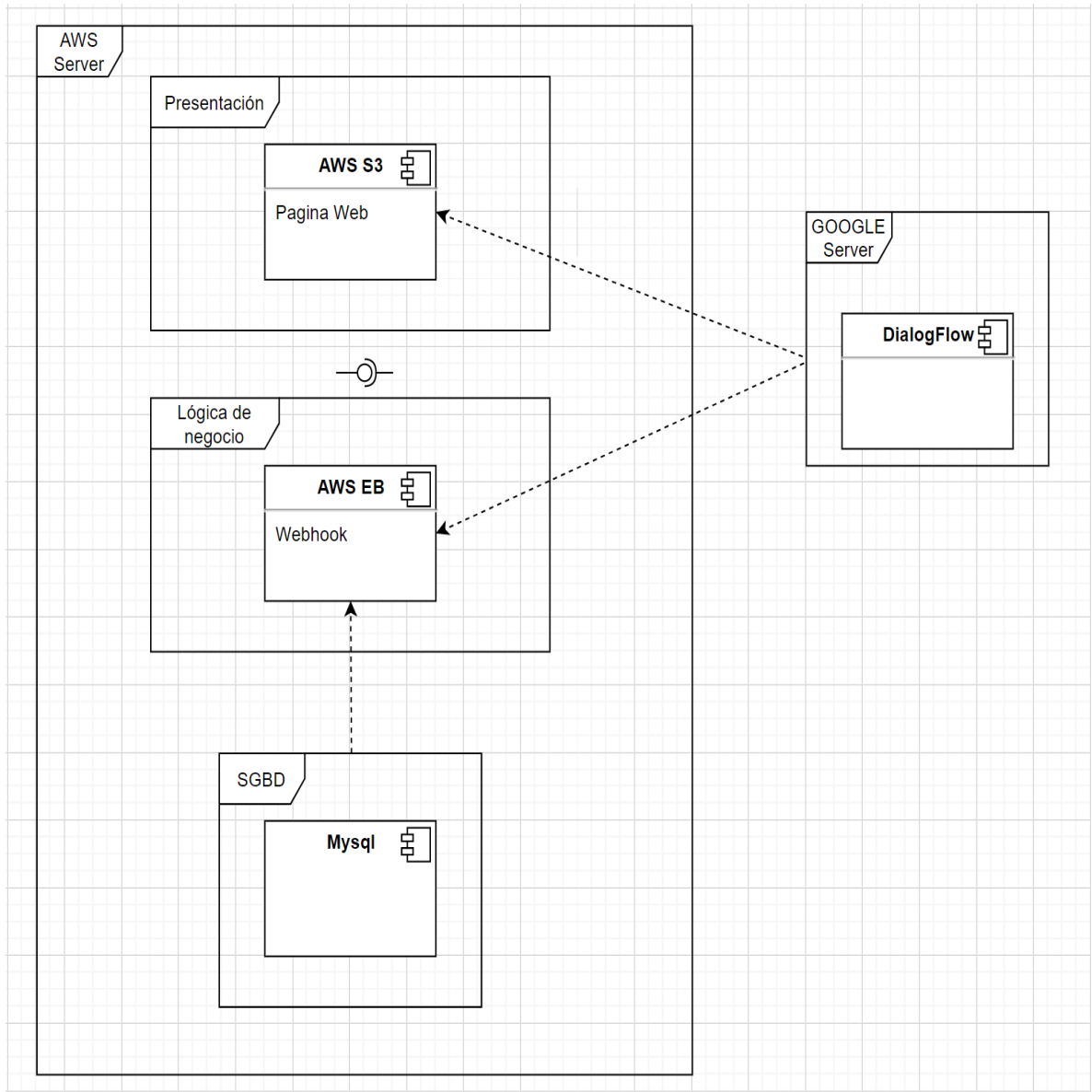


Figura 4. Componentes del sistema. (Autoría propia).

Cada componente es independiente manteniendo un bajo acoplamiento en el sistema; el webhook fue desarrollado en el lenguaje php permitiendo la comunicación entre el entorno

de Dialogflow con la base de datos; se destaca que tanto el entorno de servicios de AWS como github brindan los servicios de logs en los procesos de integración continúa permitiendo hacer un monitoreo permanente de los procesos internos, así como su estado de disponibilidad y el consumo registrado. A través de las políticas de acceso de cada componente, se manejó un control para su intercomunicación asegurando la mitigación de riesgos que se pudieran presentar por ataques al sistema.

8.3.1 Diagrama de despliegue

Se muestra la disposición de las particiones físicas, contemplando que el despliegue se realiza bajo un entorno de integración continua que involucra 2 procesos principales:

- Configuración de servidor aws: este proceso se ejecutó en 3 fases, primeramente, se desplego la base de datos mysql en el servicio de base de datos RDS, seguidamente se desplego el servicio bucket s3 marcando configuración de página web estática; posteriormente se configuro el servicio Elastic Beanstalk. Finalmente se configuro el servicio Codepipeline el cual ejecuta un despliegue automatizado del webhook que comunica Dialogflow con el sistema.
- Configuración de servidor de repositorio y despliegue: se configuró servidor de repositorio

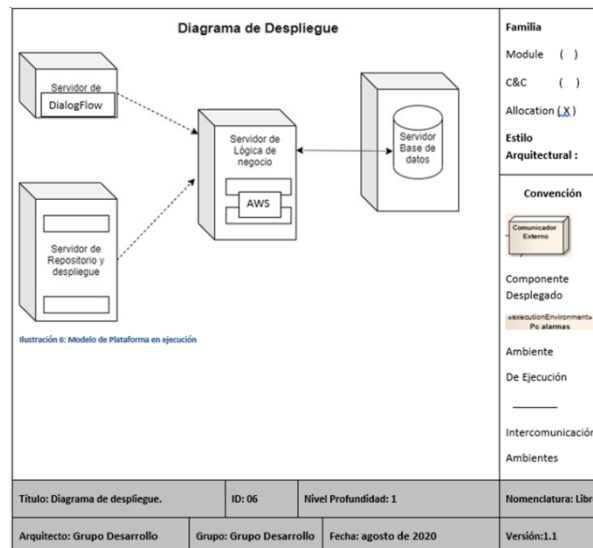


Figura 5. Diagrama de despliegue. (Autoría propia).

8.3.2 Casos de uso arquitecturalmente relevante

El enfoque del siguiente caso de uso se encuentra en fundamentado en las acciones del sistema desde el punto de vista del usuario, teniendo en cuenta que se realiza una intención describiendo los síntomas y el sistema evalúa las patologías definidas previamente y emite una respuesta final del estado de triage y las recomendaciones de dirigirse a un centro hospitalario.

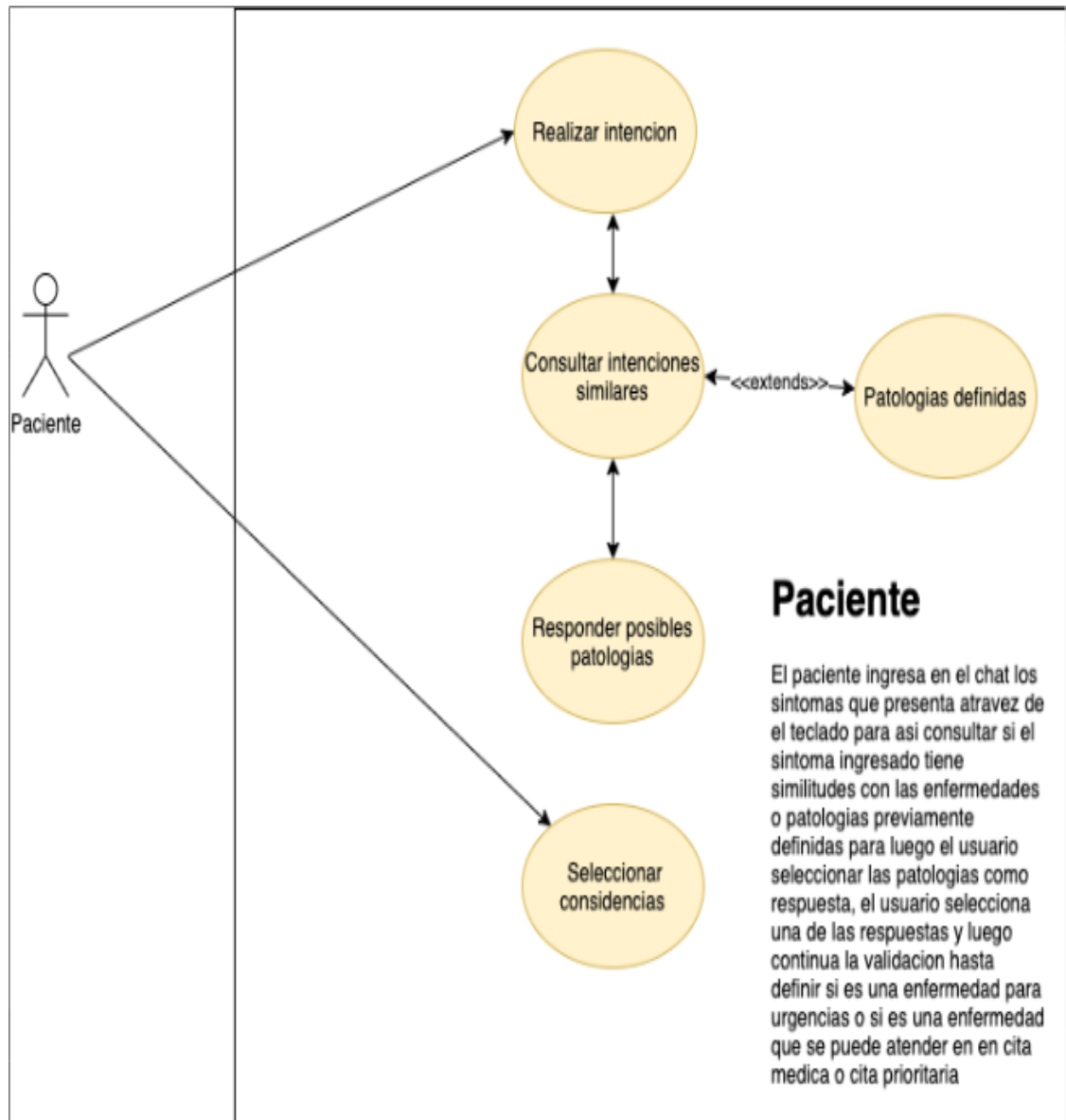


Figura 6. Diagrama usuario final. (Autoría propia).

8.3.3 Diagrama de secuencia

El presente diagrama se centra específicamente en líneas de vida de los procesos y objetos; el flujo contemplado inicia cuando el usuario ingresa una palabra en el chatbot es viaja hasta Dialogflow teniendo en cuenta que de tener la respuesta la devuelve inmediatamente o si se requiere traer datos de la base de datos, se ejecuta la consulta la cual se devuelve posteriormente como una respuesta final al usuario.

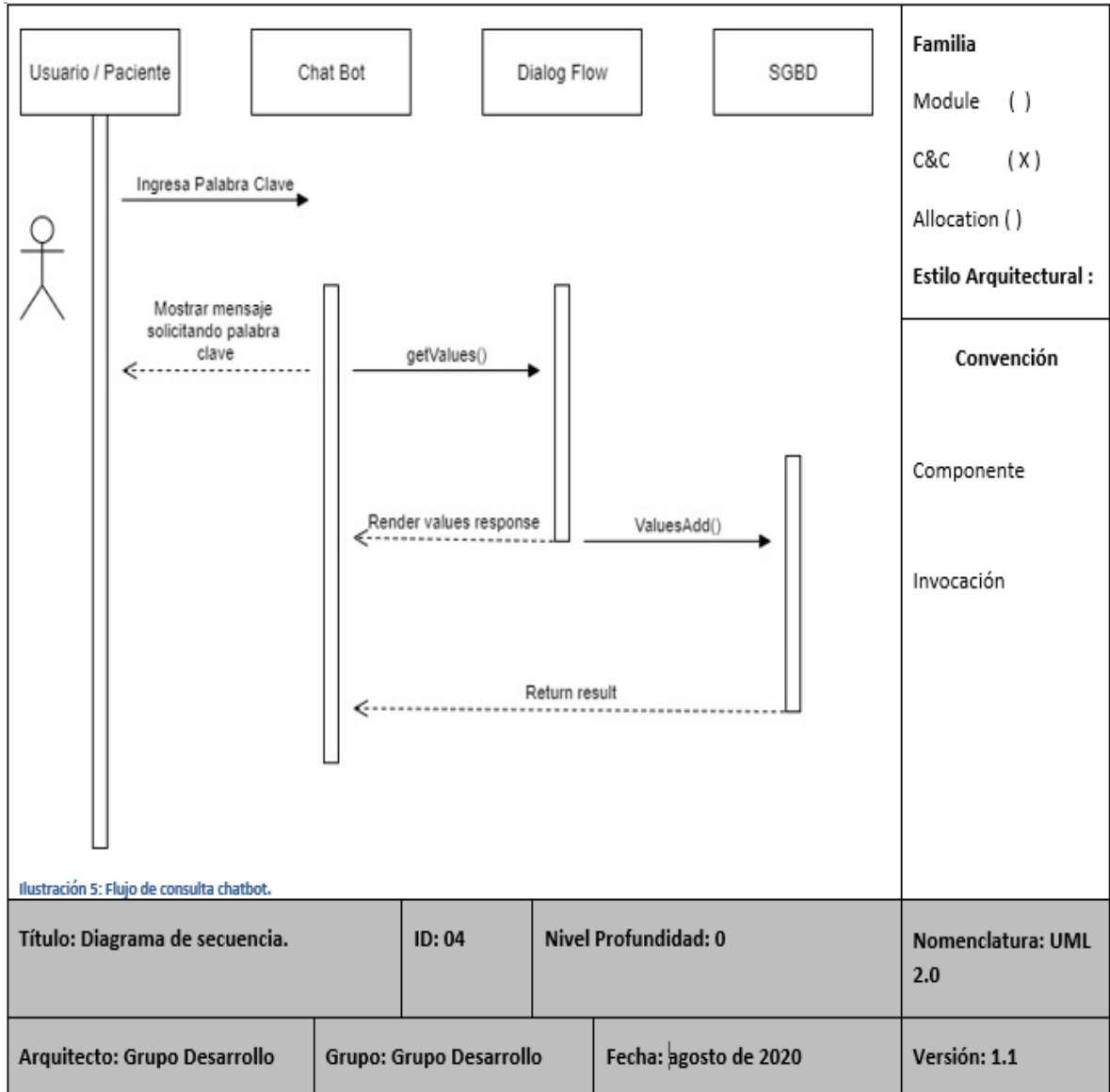


Figura 7. Diagrama de secuencia. (Autoría propia).

8.3.4 Diagrama de clases

De acuerdo a la naturaleza del proyecto este diagrama se encuentra focalizado en una estructura de clase orientada a la comunicación con Dialogflow; Las clases utilizadas en el proyecto se pueden evidenciar a través del siguiente diagrama:

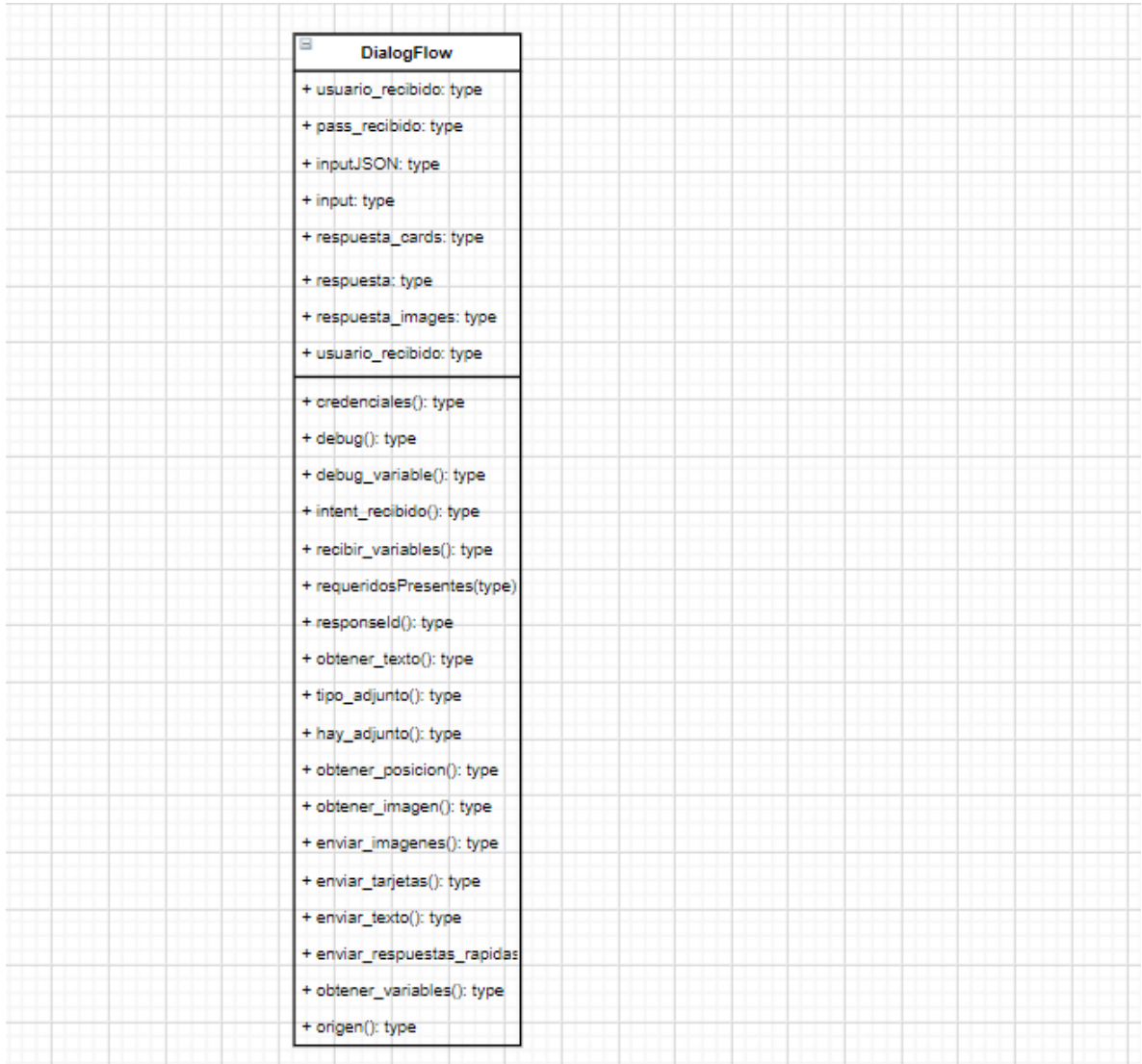


Figura 8. Diagrama de clases. (Autoría propia).

8.3.5 Arquitectura de alto nivel

Se considero una vista de nivel superior de los componentes del sistema donde visualiza la infraestructura en AWS, iniciando cuando el usuario hace una petición a través de chatbot, dicha petición pasa directamente a Dialogflow este a su vez si contiene la respuesta emite un mensaje al usuario, de lo contrario si se requiere buscar información en la base de datos, se comunica con el webhook el cual realiza la petición a la base de datos y devuelve una respuesta a dialogflow que a su vez retorna una respuesta al usuario.

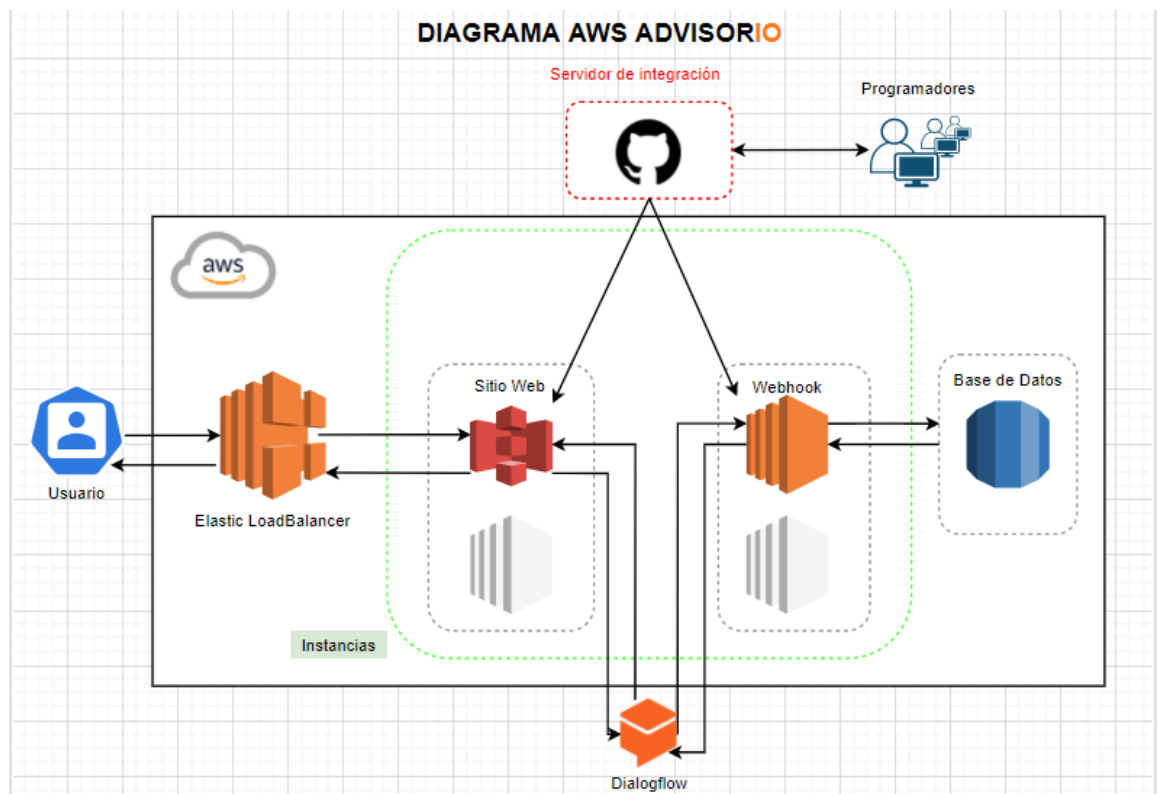


Figura 9. Arquitectura de alto nivel. (Autoría propia).

8.4 Construcción

En la fase de desarrollo se trabajó en la creación de los artefactos indicados en los diagramas teniendo en cuenta los procesos asociados a la metodología XP.

8.4.1 Creación de los repositorios de código fuente

Como se estableció en la fase de diseño, se procedió a crear dos repositorios en Github para almacenar el código fuente versionado.

- Proyecto-software-webhook-advisor: Almacena el webhook.
- Proyecto-software-chatbot-advisor: Almacena la página web.

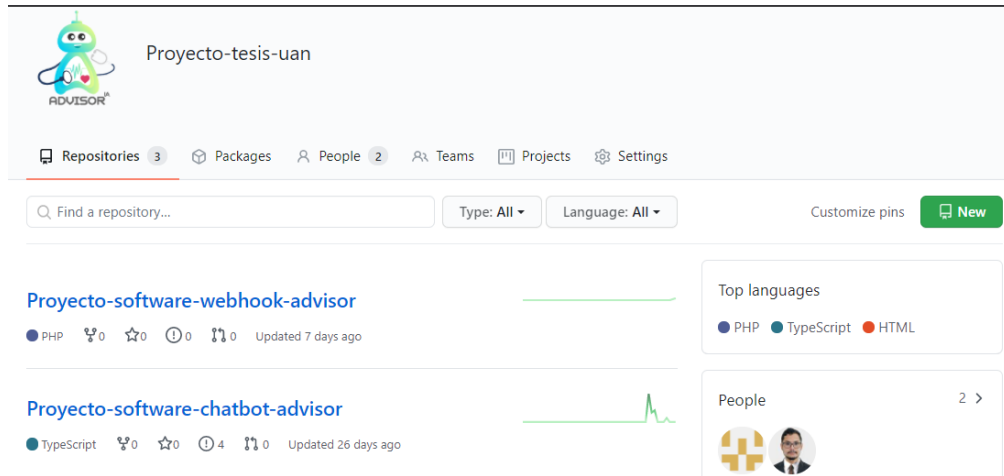


Figura 10. Repositorios proyecto Advisor. (Autoría propia).

8.4.2 Desarrollo de agente DialogFlow

Dialogflow mantiene un sistema en la lógica para la toma de decisiones basado en inteligencia artificial, por tratarse del core de la aplicación tercerizada no se deja expuestos los algoritmos que ejecutan en dichas decisiones; Es por ello que se establece una interfaz que permite la creación de parámetros de entrada, los cuales dan forma a solo una parte de la estructura del proceso interno que maneja el agente, permitiendo así poder identificar las frases claves denominadas “Intent” que a su vez establecen contextos de conversación que no son escenarios de conversaciones, estos escenarios los identifica el agente para poder dar continuidad a una conversación y evitar que frases parecidas activen respuestas que no aplican al escenario o contexto al cual se le pregunta al bot.

La plataforma DialogFlow nos permitió definir Intents, contextos, y conexión a datos externos a través de la opción fulfillment.

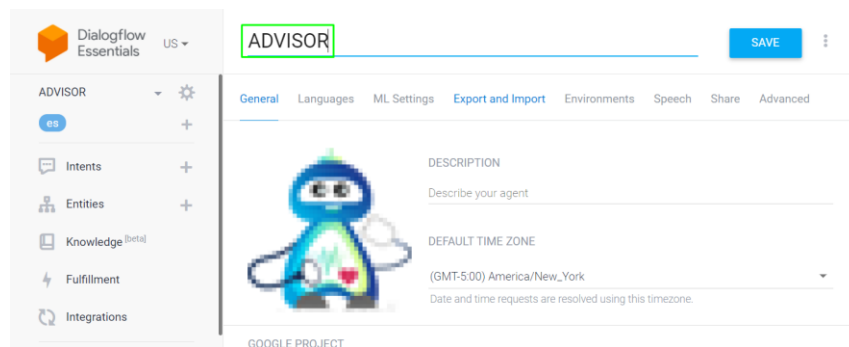


Figura 11. Agente Dialogflow. (Autoría propia).

- Desarrollo de intents

Los intent definen y clasifican la intención de usuario al momento de comunicarse con el chatbot, es decir para cada agente se establecen varios posibles intents, que en conjunto pueden establecer una conversación completa; cuando un usuario escribe una frase en el chatbot está comparando esas palabras con los intents creados en el agente permitiendo así dar una respuesta acertada.

- Desarrollo de contextos

Para controlar la conversación se utilizan los denominados “contextos”, es una de las partes más importantes para correlacionar lo que se está hablando actualmente con lo que ya se ha hablado, pudiendo acceder a de esta manera a datos previamente capturados en la conversación; Para ello se utilizan contextos de entrada y salida que se definen a continuación:

Los contextos de salida permiten controlar todos los contextos activos; Es decir cuando existe una coincidencia con un intent, automáticamente se activa el contexto, al que se puede acceder posteriormente desde otro intent.

Los contextos de entrada permiten dar continuidad y obtener información de un intent anterior que tiene relación con el intent actual; es decir coloca en contexto de la conversación al chatbot.

- Integración a fulfillment

El proceso de integración con el webhook fue dado mediante la plataforma Dialogflow a través de la opción fulfillment, requiriendo un usuario y una contraseña para el acceso.

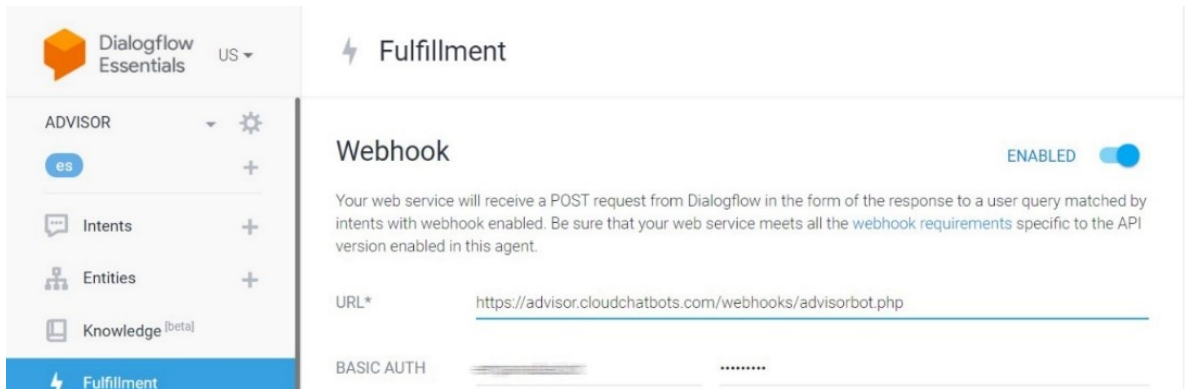


Figura 12. Integración Dialogflow Fulfillment. (Autoría propia).

8.4.3 Desarrollo de la base de datos

El esquema contemplado para la base de datos en el proyecto incluye una tabla donde serán almacenados los centros hospitalarios que serán presentados a los usuarios a través del chatbot.

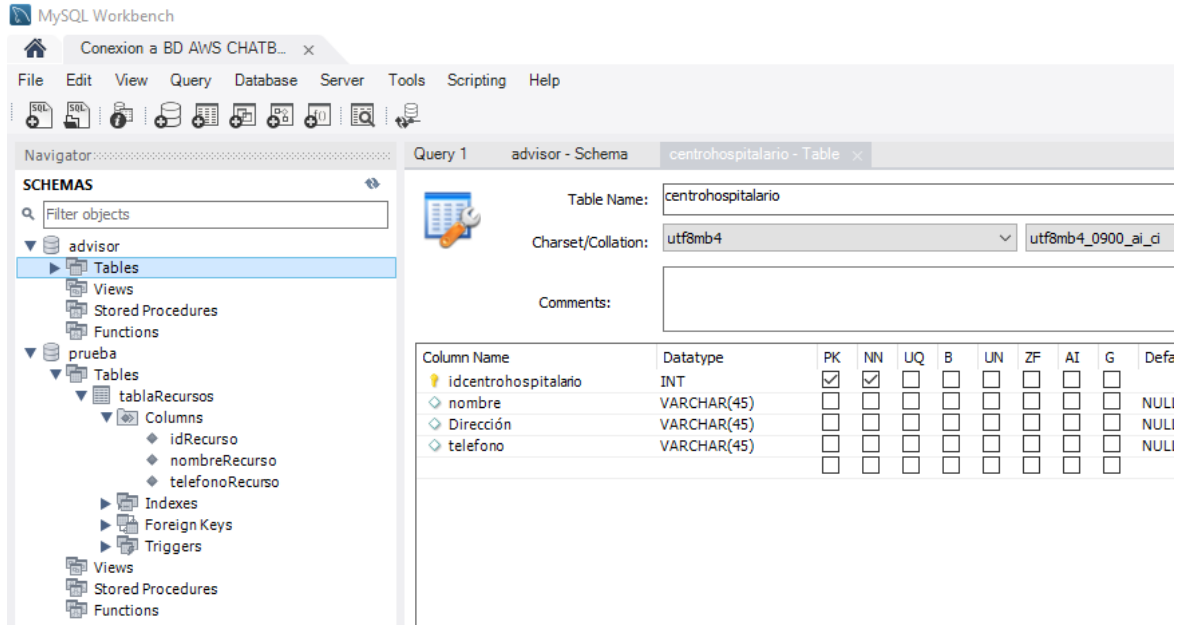


Figura 13. Base de datos. (Autoría propia).

La estructura de la base de datos se encuentra conformada de la siguiente manera:

centrohospitalario
+ field: idcentrohospitalario
+ field: nombre
+ field: direccion
+ field: Telefono

Figura 14. Estructura tabla centro hospitalario. (Autoría propia).

8.4.4 Desarrollo webhook

El desarrollo del webhook se realizó bajo el lenguaje de programación PHP; implementando una librería externa con métodos que permiten la comunicación tanto con la base de datos como con Dialogflow.

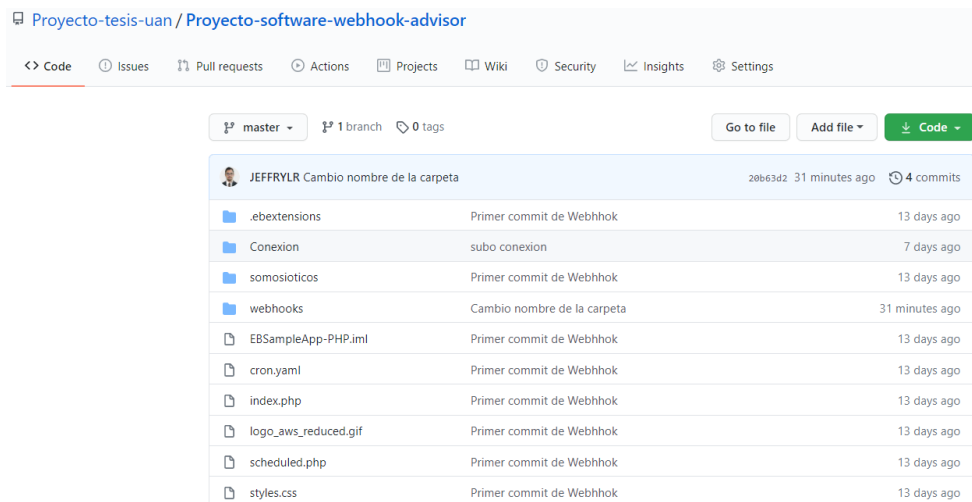


Figura 15. Repositorio webhook. (Autoría propia).

- Librería somosioticos_dialogflow

Para facilitar el desarrollo y permitir el desacoplamiento del código fuente se utilizo la librería somosioticos [x], la cual contiene métodos que facilitan el flujo de trabajo con Dialogflow.

Tabla 11

Método Librería SomosIoticos

Métodos de la Librería Somosioticos	
Reciben parámetros	Envían Parámetros
credenciales()	function enviar_tarjetas(\$tarjetas, \$plataforma)
function intent_recibido()	function enviar_texto(\$texto)
recibir_variables(\$nombre))	function enviar_respuestas_rapidas(\$respuestas, \$plataforma)
function responseId()	
function obtener_texto()	
function obtener_variables()	

Nota. Esta tabla especifica los métodos de la librería SomosIoticos (Autoría propia).

8.5 Pruebas

El desarrollo de pruebas se realizó teniendo en cuenta los conceptos de pruebas unitarias, así como las pruebas de código estático:

8.5.1 Pruebas de código estático

Estas pruebas se realizaron mediante la integración de la herramienta SonarCloud, la cual permite la detección de errores en el código de manera estática, generando métricas en tiempo real y activándose cuando se realiza una modificación sobre la rama del proyecto.

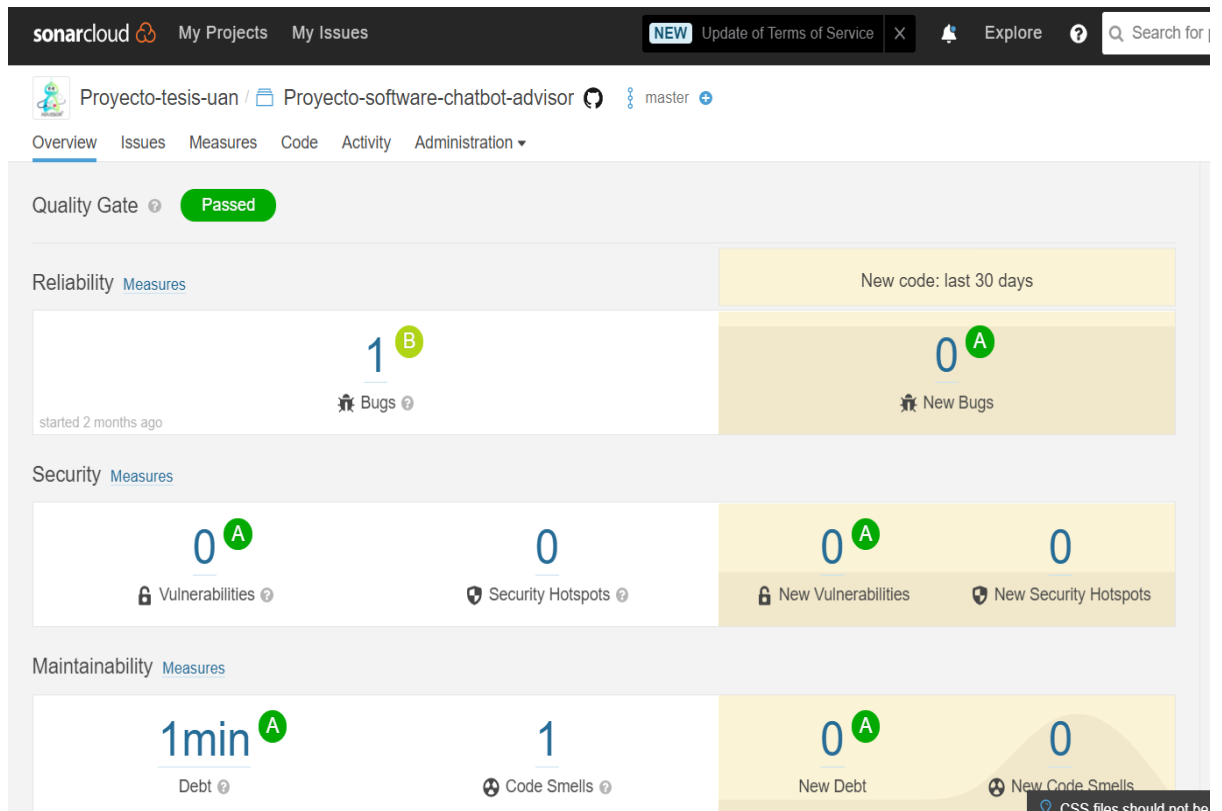


Figura 16. Herramienta de análisis estático SonarCloud. (Autoría propia).

8.5.2 Pruebas Unitarias

Las pruebas unitarias se desarrollaron los componentes de angular (Karma, Protactor, Jasmine), configurándose un archivo de test y este se ejecuta en el pipeline desarrollado en github.

8.5.3 Pruebas de carga

Para las pruebas de carga o concurrencia se utilizó el software Apache JMeter teniendo en cuenta una carga de 200 peticiones por segundo, resultando con status 200 como se muestra en la siguiente imagen:

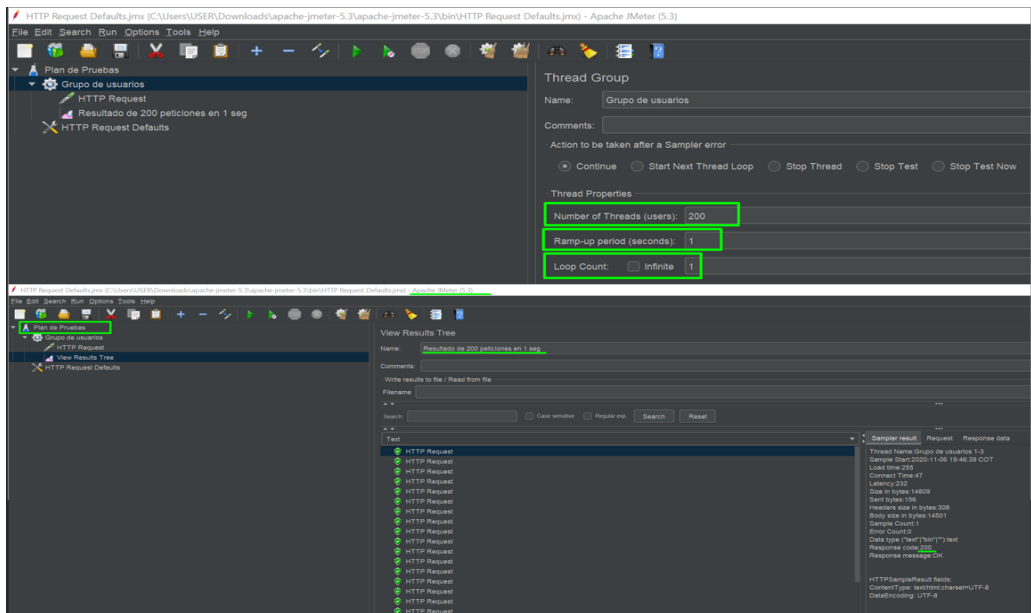


Figura 17. Entorno de prueba JMeter. (Autoría propia).

8.6 Instalación y configuración

El proceso de instalación, configuración y despliegue se realizó en 2 etapas, bajo un contexto de integración continua y despliegue continuo.

8.6.1 Despliegue de aplicación

Mediante las acciones ofrecidas por Github Actions se creó un pipeline el cual se ejecuta inmediatamente después de subir el código. Este proceso se encuentra definido en un archivo (.yaml) el cual debe estar contenido en la carpeta asignada por Github denominada workflow.

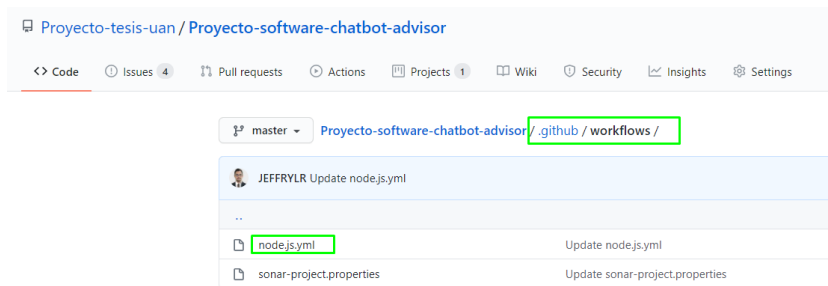


Figura 18. Carpeta workflow Github. (Figura autónoma).

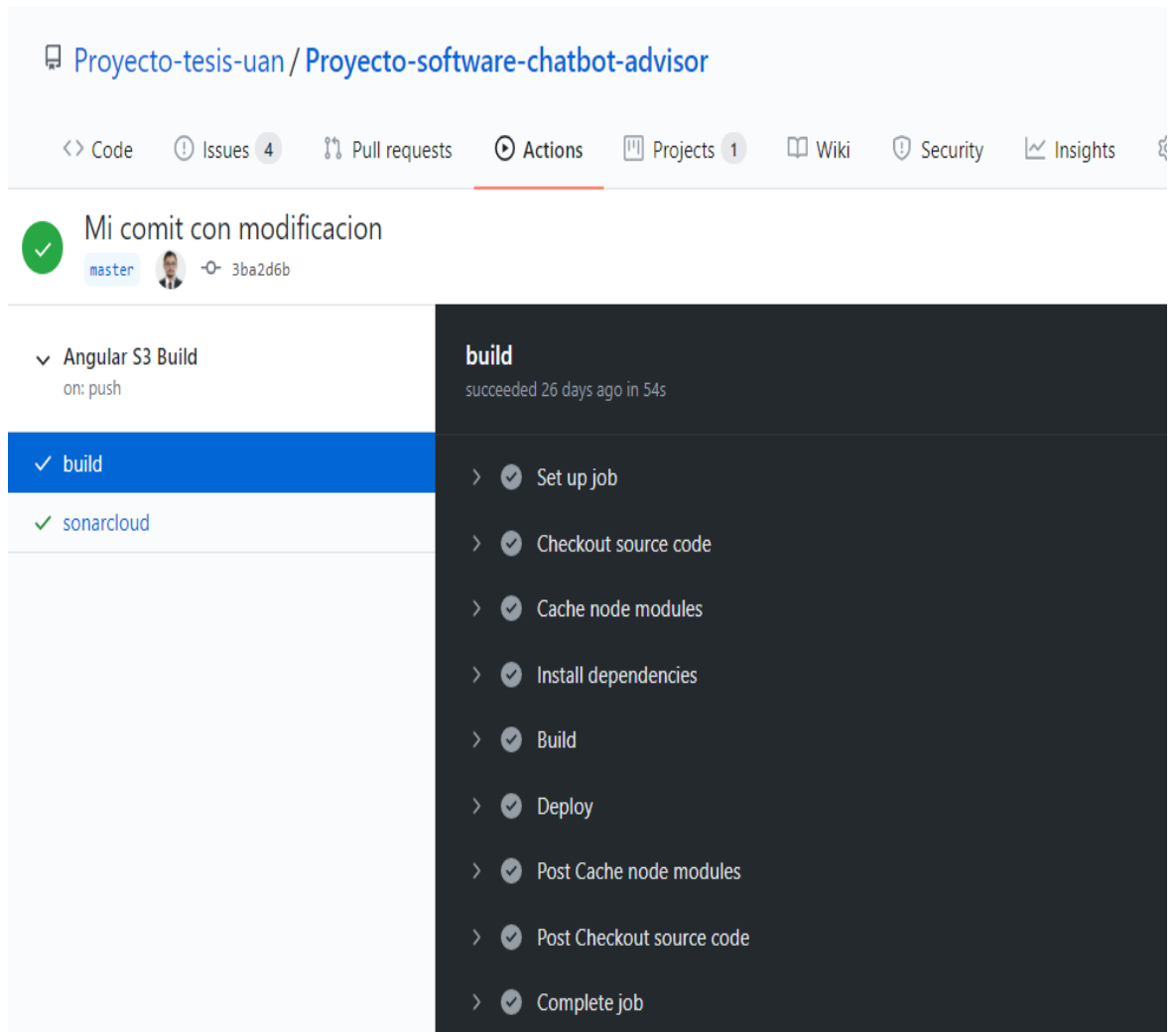


Figura 19. Procesos de despliegue Github Actions. (Autoría propia).

El proceso concreto que se ejecuta en el pipeline, copia los archivos del repositorio github hasta un bucket s3 de AWS, y desde ese punto entra en producción.

A nivel de seguridad se tuvieron contemplados los llamados “Secretos” en Github, los cuales permiten la integración con herramientas como sonar y adicionalmente la integración con aws, garantizando la seguridad en los procesos de comunicación entre los servicios.

Posteriormente a la ejecución del pipeline, podemos visualizar a continuación la interfaz grafica de la aplicación, con un entorno grafico amigable e intuitivo de una sola pagina y con el chatbot Advisor a la vista.

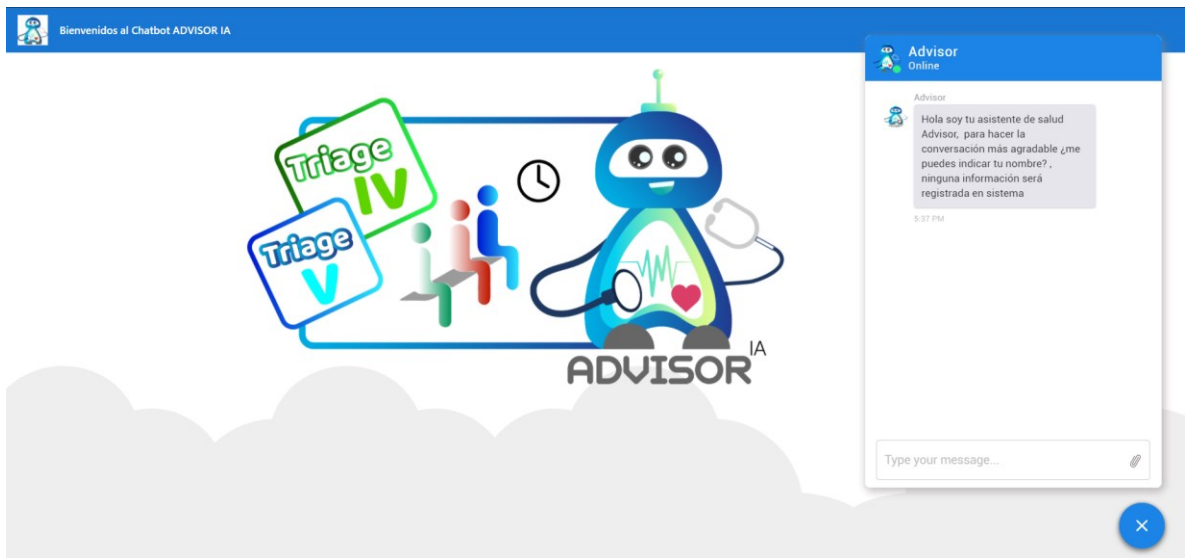


Figura 20. Aplicación final. (Autoría propia).

8.6.2 Despliegue de webhook

El despliegue del webhook se realiza desde el servicio de AWS llamado Codepipeline, que permitió realizar la configuración del pipeline o canalización del proceso.

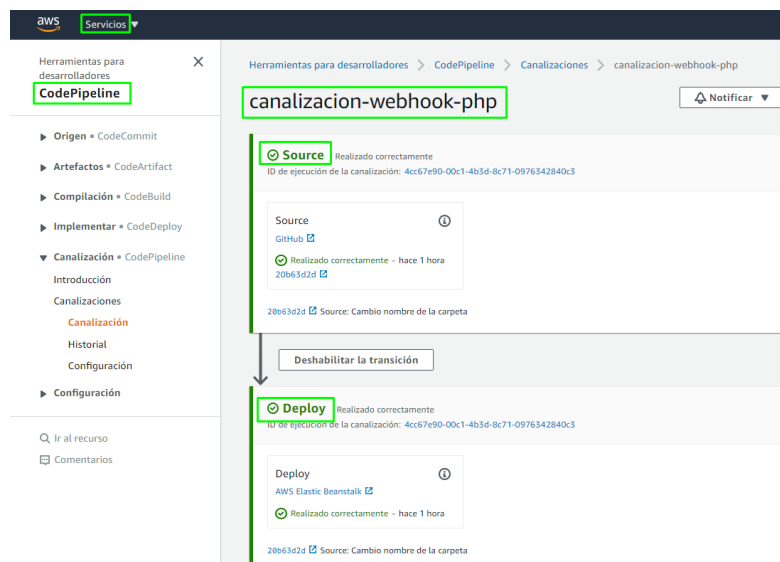


Figura 21. Procesos de despliegue Webhook. (Autoría propia).

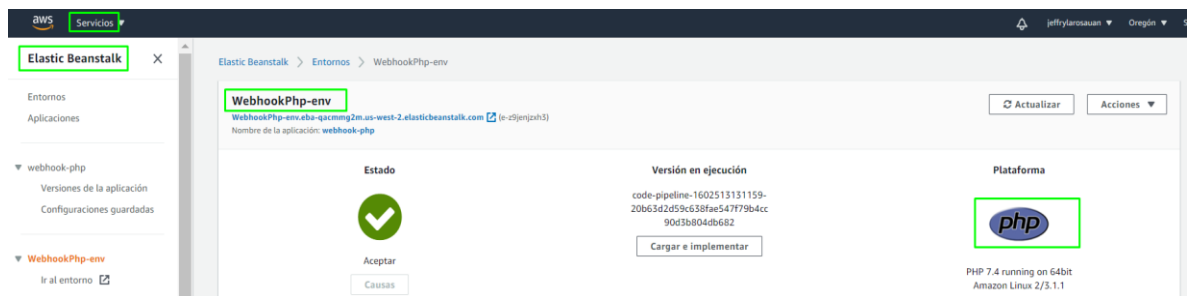


Figura 22. Entorno Elastic Beanstalk. (Autoría propia).

8.7 Aspectos de seguridad en la nube

Cada elemento de sistema cuenta con un componente de seguridad dependiendo del entorno donde se encuentre cumpliendo la labor; en el caso AWS, este entorno cuenta con un artefacto de acceso de seguridad denominado IAM (Identity and Access Management) ANEXO B - *Servicios de seguridad AWS*, donde se crean las políticas de los permisos y roles asociados a cada servicio en la nube AWS.

A nivel de Google, Dialogflow cuenta con la opción de establecer un token de seguridad para los webhooks, lo que permite que el webhok tenga la restricción de acceso a solicitud de peticiones. Teniendo en cuenta que los repositorios son focos comunes de ataques, al repositorio principal del código fuente Github, se le configuraron los accesos por factor de doble autenticación, y se establecieron Tokens de seguridad para la comunicación con el analizados de código estático Sonalcloud y el despliegue de AWS.

9.2 ANEXO B- Servicios de seguridad AWS
























Servicios de seguridad, identidad y conformidad de AWS		
Categoría	Casos de uso	Servicio de AWS
Identity & Access Management	Administre de manera segura el acceso a los servicios y los recursos	 AWS Identity & Access Management (IAM)
	Servicio de inicio de sesión único (SSO) en la nube	 AWS Single Sign-On
	Administración de identidades para las aplicaciones	 Amazon Cognito
	Microsoft Active Directory administrado	 AWS Directory Service
	Servicio simple y seguro para compartir recursos de AWS	 AWS Resource Access Manager
	Gobernanza y administración centralizadas en cuentas de AWS	 AWS Organizations
Detección	Centro unificado de seguridad y conformidad	 AWS Security Hub
	Servicio administrado de detección de amenazas	 Amazon GuardDuty
	Analice la seguridad de las aplicaciones	 Amazon Inspector
	Registre y evalúe las configuraciones de sus recursos de AWS	 AWS Config
	Realice un seguimiento de la actividad de los usuarios y el uso de las API	 AWS CloudTrail
	Administración de la seguridad para dispositivos compatibles con IoT	 AWS IoT Device Defender
Protección de infraestructuras	Protección frente a ataques DDoS	 AWS Shield
	Filtre el tráfico web malintencionado	 AWS Web Application Firewall (WAF)
	Administración central de reglas de Firewall	 AWS Firewall Manager
Protección de datos	Descubra y proteja sus datos confidenciales a escala	 Amazon Macie
	Administración y almacenamiento clave	 AWS Key Management Service (KMS)
	Almacenamiento de claves en hardware a efectos de conformidad normativa	 AWS CloudHSM
	Aprovisionamiento, administración e implementación de certificados públicos y privados SSL/TLS	 AWS Certificate Manager
	Altere, administre y recupere datos confidenciales	 AWS Secrets Manager
Respuesta frente a incidencias	Investigue los posibles problemas de seguridad	 Amazon Detective
	Recuperación ante desastres rápida, automatizada y rentable	 CloudEndure Disaster Recovery
Conformidad	Portal gratuito autoservicio para el acceso bajo demanda a los informes de conformidad de AWS	 AWS Artifact

Figura 24. Servicios de seguridad AWS. (Figura autónoma).

10. CONCLUSIONES

Se concluye que a través de la utilización de la ingeniería de software aplicada a el desarrollo de un Chatbot (Advisor) que implementa inteligencia artificial, se genera un alto valor al concepto de atención inmediata en la valorización del estado de triage de un paciente asociado a un estado de emergencia. Permitiendo así que los tiempos de espera disminuyan considerablemente, factor el cual afecta a gran número de instituciones hospitalarias en Colombia.

El software realizado permite la reducción hasta de 50% del tiempo en que un paciente debe esperar para ser atendido, garantizando la entrega de servicio continua a través de una plataforma web, y accesible desde dispositivos como celulares, tabletas y computadores, las 24 horas del día.

A nivel de software se contempló que la infraestructura soporte el escalamiento de la aplicación, así como la independencia de componentes; Así también garantizando que las estructuras de dialogo del chatbot se respaldan en un formato que puede ser utilizado en otras aplicaciones asegurando el desacoplamiento de una tecnología Dialogflow en particular.

11. REFERENCIAS-BIBLIOGRAFIA

- [1] Coloquio, “La crisis de la salud en Colombia.”
- [2] D. A. Mendoza Casseres, M. González Conde, R. A. Corcho Martínez, and A. Berdugo Alonso, “Aplicación de la simulación discreta en el área de urgencias de una institución prestadora de servicios para disminuir pérdida de pacientes,” *Rev. Fac. Ing. la Univ. Libr. Secc. Barranquilla*, vol. 21, pp. 55–71, 2016.
- [3] Universidad colegio mayor de nuestra señora del rosario, “Congestión en el servicio de urgencias ¿retrasa la primera dosis de antibiótico o analgésico?.” .
- [4] R. Bogotá, “Hombre falleció en urgencias de hospital de Bogotá y horas después lo llamaron para atenderlo,” *El Espectador*, 2015
- [5] K. y P. A. Middleton, Katherine; Butt, Mobasher; Hammerla, Nils; Hamblin, Steven; Mehta, “Sorting out symptoms: design and evaluation of the „babylon check“ automated triage system,” vol. abs/1606.0, 2016.
- [6] Mediktor, “Mediktor: respuesta profesional a problemas de salud desde el primer síntoma,” *SMART Heal.*, 2016.
- [7] Mediktor, “Mediktor,” Mediktor, 2017. [Online]. Available: Mediktor.
- [8] W. Soler, M. Gomez Munoz, E. Bragulat, and A. Alvarez, “El triaje: herramienta fundamental en urgencias y emergencias. *Anales sis San Navarra*,” *Scielo*, vol. 33 Suppl.1, no. 1137–6627, pp. 55–68, 2010.
- [9] iTriage LLC, “iTriage - Health, Doctor, Symptoms and Healthcare search on the App Store.”
- [10] I. JXT Applications, “Quick Triage App - Aplicaciones Android en Google Play.”
- [11] E. Peris, ““Triage Territorial - Aplicaciones Android en Google Play.”
- [12] START Free, “START Free - Aplicaciones Android en Google Play.”

- [13] prn education, “CTAS - Triage - OFFICIAL.” 2014
- [14] I. Learning, “Fast Triage App Lite,” 2015.
- [15] I. Publishing, “EMS ACLS Guide - Aplicaciones Android en Google Play.” 2013.
- [16] A. Álvarez, Luisa; Pinedo, Andrés; Rocha, “No Title,” Pontificia Universidad Javeriana, 2016.
- [17] x-rol, “Telemedicina.” [Online]. Available: <http://www.x-rol.com/caracteristicas.html>.
- [18] D. Acuña, B. Lara, C. Basaure, Ó. Navea, C. Kripper, and F. Saldías, “Estratificación de riesgo (Triage) en el Servicio de Urgencia,” *Rev. Chil. Med. Intensiva*, vol. 30, no. 2, pp. 79–86, 2015
- [19] L. F. Correa-Serna and J. Pedraza, *Guías de manejo de urgencias*, vol. 3. 2009.
- [20] https://github.com/ioticos/simple-dialogflow-php-library/blob/master/somosioticos_dialogflow.php