



Universidad
Nacional
de Loja

Facultad de Energía, las Industrias y los Recursos Naturales no Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

**Sistema de información sobre COVID-19 mediante
llamadas telefónicas con la ayuda de un agente
inteligente**

**Information system for COVID-19 by means of
telephone calls with the help of an intelligent agent**
Desarrollo de Software

PROYECTO DE TITULACIÓN EN
INGENIERÍA EN SISTEMAS.

Autor:

◇ ORCID, Sandro Michael Córdova Carrión, sandro.cordova@unl.edu.ec

Tutor:

- José Oswaldo Guamán Quinche, Mg.Sc.

Cotutor:

- José Oswaldo Guamán Quinche, Mg.Sc.



LOJA - ECUADOR
2020

Certificación

Certificación de Autoría del Proyecto

Yo Sandro Michael Córdova Carrión, estudiante de la Universidad Nacional de Loja, declaro en forma libre y voluntaria que el presente proyecto de trabajo de titulación que versa sobre **Sistema de información sobre COVID-19 mediante llamadas telefónicas con la ayuda de un agente inteligente - Information system for COVID-19 by means of telephone calls with the help of an intelligent agent**, así como la expresiones vertidas en la misma son autoría del compareciente, quien ha realizado en base a recopilación bibliográfica primaria y secundaria. En consecuencia asumo la responsabilidad de la originalidad de la misma y el cuidado al remitirse a las fuentes bibliográficas respectivas para fundamentar el contenido expuesto.

Atentamente,
Sandro Michael Córdova Carrión

Agradecimiento

Dedicatoria

Cesión de derechos

Índice general

Certificación	i
Agradecimiento	iii
Dedicatoria	iv
Cesión de derechos	v
1. Resumen	2
2. Summary	3
3. Introducción	4
4. Objetivos	5
4.1. General	5
4.2. Específicos	5
5. Marco Teórico	6
5.1. Fundamentación teórica	6
5.1.1. Inteligencia artificial	6
5.1.2. Agente inteligente	6
5.1.3. Características de los agentes	6
5.1.4. Ejemplos de tipos de Agentes	7
5.1.5. Clases de agentes inteligentes	8
5.1.6. Métodos para el desarrollo y construcción de los agentes inteligentes	8
5.1.7. Metodologías para el desarrollo de agentes inteligentes	9
5.1.8. Arquitectura de los agentes inteligentes	10
5.1.9. Arquitecturas deliberativas	10
5.1.10. Arquitecturas reactivas	10
5.1.11. Arquitecturas híbridas	10
5.1.11.1. Capa reactiva	11
5.1.11.2. Capa de conocimiento	11
5.1.11.3. Capa social	11
5.1.12. Sistemas multiagente	11
5.1.13. Comunicación entre agentes	12

5.1.14. ABLE	12
5.1.15. JADE	13
5.1.16. Metodologías de desarrollo	13
5.1.17. Selección de la metodología de desarrollo de software	14
5.1.18. Selección de la metodología de desarrollo ágil	14
5.1.19. Metodología de desarrollo SCRUM	14
5.1.20. Metodología de desarrollo Kendall y Kendall	17
5.1.21. Medidas de bioseguridad	18
6. Materiales y métodos	19
6.1. Contexto	19
6.2. Proceso	19
6.3. Recursos	20
6.3.1. Recursos Científicos	20
6.3.1.1. Métodos	20
6.3.1.2. Técnicas	20
6.3.2. Recursos Académicos	22
6.3.3. Recursos Técnicos	22
6.3.4. Recursos Éticos	22
6.4. Participantes	23
7. Resultados	24
7.1. Análisis de metodologías para el desarrollo de software	24
7.2. Entrevista para la elicitación de requisitos	24
7.3. Historias de usuario	24
7.4. Requisitos funcionales	25
7.5. Requisitos no funcionales	25
7.6. Metodología para el desarrollo de software	25
8. Conclusiones	28
9. Recomendaciones	29
Bibliografía	30
A. Anexo I	32
B. Anexo II	45
C. Anexo III	64
D. Anexo IV	82

Índice de figuras

Índice de tablas

5.1. Metodologías de desarrollo de software	15
5.2. Metodologías de desarrollo de software ágiles	16
7.1. Requisitos preliminares.	25
7.2. Historias de usuario.	26
7.3. Requisitos funcionales.	27
7.4. Requisitos no funcionales.	27

**Sistema de información sobre
COVID-19 mediante llamadas
telefónicas con la ayuda de un
agente inteligente**
Desarrollo de Software

1. Resumen

2. Summary

3. Introducción

4. Objetivos

4.1. General

Desarrollo de un prototipo funcional de ayuda al usuario para un sistema de llamadas telefónicas de un centro de salud basado en un agente inteligente.

Development of a functional prototype of user assistance for a health center telephone call system based on an intelligent agent.

4.2. Específicos

- Diseñar el módulo de software del agente inteligente para el sistema de llamadas telefónicas utilizando el lenguaje de programación Java en el framework JADE y la metodología de desarrollo Kendall.
Design the intelligent agent software module for telephone calls system using Java programming language in JADE framework and Kendall development methodology.
- Implementar el módulo de software del agente inteligente al sistema de llamadas telefónicas.
Implement the intelligent agent software module to the telephone calls system.
- Evaluar la aceptación y correcto funcionamiento del módulo de software en el sistema de llamadas telefónicas.
Assess the acceptance and correct functioning of the software module in the telephone calls system.

5. Marco Teórico

5.1. Fundamentación teórica

5.1.1. Inteligencia artificial

En el mundo en que vivimos en la actualidad casi todo lo que fabrica el hombre a nivel tecnológico tiene Inteligencia Artificial como: los juegos de azar, los videojuegos, los robots, aparatos médicos, electrodomésticos, son algunos de los dispositivos donde se hace uso de esta ciencia. Pero conforme se crean estos nuevos sistemas tecnológicos, se debe innovar y crear nuevas cosas que sean de gran utilidad para el hombre, para las empresas, las industrias, las instituciones educativas, entre otras.[1] Es por ello que, la tecnología se encuentra en constante evolución, siendo el internet uno de los más grandes e importantes inventos, donde la inteligencia artificial es parte fundamental como rama de la ciencia computacional.

5.1.2. Agente inteligente

Un agente inteligente se puede considerar como una entidad capaz de percibir su entorno, tener autonomía y de retroalimentar su conocimiento. En las plataformas se encarga de percibir el nivel de inteligencia y las dificultades que presenta el usuario, obteniendo un perfil de usuario de acuerdo a lo que percibe, con la finalidad de poder brindar soluciones de acuerdo a las necesidades del individuo.[1] Los agentes inteligentes cumplen con algunas características de percepción, acción, metas, ambiente y tienen propiedades de acuerdo al ambiente en que se desarrollan.

5.1.3. Características de los agentes

Las siguientes características de agentes fueron obtenidas de: B. Lopez [1]

- Reactivo: el agente tiene la capacidad de responder a cambios en el entorno donde se encuentra situado.
- Proactivo: el agente trata de cumplir su propio plan u objetivo.
- Social: el agente debe poder comunicarse con otros agentes por medio de cualquier lenguaje.

5.1.4. Ejemplos de tipos de Agentes

Las características de los siguientes tipos de agentes fueron obtenidas de: B. Lopez [1]

- Sistema de diagnóstico médico
 - Este sistema tiene como rendimiento o meta la reducción al mínimo de los costos y de las demandas, manteniendo al paciente saludable.
 - El entorno donde se trabaja es en el hospital donde se encuentran el paciente y el personal como las enfermeras y los doctores.
 - Los actuadores o acciones que se realizan son: preguntas, pruebas, tratamientos, diagnósticos, casos, impresora, monitor entre otras.
 - Los sensores o percepciones en este sistema son el síntoma, evidencias y respuestas del paciente. Teclado para emitir dicha información.
 - Sistema para el análisis de imágenes satelitales
 - Su meta es la clasificación correcta de la imagen
 - El entorno son las imágenes enviadas desde un satélite en órbita, conexión con el satélite en órbita.
 - La acción en este sistema son la visualización e impresión de una clasificación de escena.
 - Los sensores que se utilizan son: pixel de intensidad y colores diversos (matriz de bits)
 - Robot clasificador (seleccionador) de partes
 - Su meta es poner las partes en el bote que les corresponde y su rendimiento es el porcentaje de piezas clasificadas
 - El entorno son las bandas transportadoras sobre la que se encuentra las partes, y los botes en los que se depositaran las partes clasificadas.
 - Los actuadores recogen las partes y las clasifican en bote. Brazos y manos articulados.
 - Los sensores que se utilizan son: píxel de intensidad variable (mapa de bits). Cámara, sensor angular.
 - Control de una refinería
 - La meta en este sistema es maximizar la seguridad, pureza y rendimiento (producción).
 - El entorno es la refinería y los operadores
-

- Las acciones son: abrir y cerrar válvulas, ajustar la temperatura, los actuadores son las válvulas, bombas, calentadores y monitores.
- La percepción es la lectura de temperatura y la presión y utiliza sensores químicos.
- Tutor interactivo de inglés
 - La meta de este sistema es que el estudiante obtenga la máxima calificación en una prueba.
 - El entorno son los grupos de estudiantes y la agencia examinadora.
 - Los actuadores que se emplean son: ejercicios en línea, impresos, de audio, visualizar correcciones y sugerencias.
 - Las percepciones son las palabras escritas a través del teclado, y emisión de voz. Los sensores en este caso son el teclado y micrófono.
- Conductor de taxi
 - Su meta consiste en ofrecer un viaje seguro, rápido, sin infracciones, cómodo, su rendimiento es la obtención máxima de ganancias.
 - El entorno que rodea a un conductor de taxi son: caminos, trafico, peatones, cliente, semáforos.

5.1.5. Clases de agentes inteligentes

Los agentes se pueden clasificar de acuerdo a la función que ejecutan y a las propiedades que tienen como: [1]

- Agentes de interfaz: tienen la capacidad de adaptarse a la forma de aprendizaje del usuario y sirve de asistente en la comprensión de nuevos temas.
- Agentes móviles: procesos computacionales capaces de viajar por nodos de una red, por computadoras, navegar por internet y efectuando intercambios de información.
- Agentes de información: sirven de ayuda para mejorar el manejo de la sobrecarga de información en la web.

5.1.6. Métodos para el desarrollo y construcción de los agentes inteligentes

- Proceso de Definición del Problema del Agente
 - Proceso de Adquisición y Representación del conocimiento.
 - Proceso de Diseño del Agente.
-

- Proceso de Implantación del Agente.
- Proceso de Prueba del Agente.

5.1.7. Metodologías para el desarrollo de agentes inteligentes

A continuación, se plantean las diferentes metodologías para el desarrollo de agentes inteligentes:

MaSE: Es una metodología basada en el análisis y diseño de sistemas multiagente, esta herramienta combina varios modelos dentro de una misma metodología con altos niveles de comprensión, facilitando la transformación de los modelos generados a código fuente para la implementación de los agentes por parte de los programadores. [2] Esta metodología cuenta con un limitante técnico, debido a que los resultados obtenidos con ella no son aplicables a todos los frameworks.

Gaia: Esta metodología permite el paso de un análisis sistemático identificado mediante requerimientos a un diseño con alto nivel de detalle que es implementado directamente. Además, de la captura de información, la fase de requerimientos se realiza a mediante paradigmas definidos directamente por la metodología, lo cual le permite un constante movimiento entre análisis e implementación de forma limpia y sin perder la esencia de las funcionalidades requeridas para los agentes inteligentes, empleando lenguajes de programación dedicados para adaptar el diseño generado por la metodología.[2]

Prometheus: Esta metodología abarca el ciclo de desarrollo de agentes inteligentes de principio a fin desde el análisis, diseño, desarrollo y la implementación, brindando a los equipos de desarrollo la facilidad para poder generar diseños conceptuales de los agentes y procesos necesarios a través de notaciones definidas por la metodología paso a paso, inspiradas en reglas simples. [2]

Esta metodología cuenta con propósitos generales, donde se asumen características de la arquitectura de los agentes en las fases iniciales, pero en el desarrollo de los proyectos se dejan a un lado las suposiciones planteadas creando ideas solidas que serán empleadas en la implementación de los sistemas multiagente.

Zeus: Sistema de desarrollo de agentes de código abierto desarrollado en el lenguaje de programación java, considerado como un kit de herramientas para la construcción de aplicaciones basadas en agentes inteligentes, su campo de aplicación se da en la fase de desarrollo del proyecto donde apoya al proceso en la planeación y la definición de las acciones de los agentes. [2] Adicionalmente, Zeus proporciona facilidades de comunicación entre sistemas multiagente empleando el protocolo de transporte TCP/IP y empleando sockets como mecanismo de entrega de mensajes.

Otra de las características principales de Zeus es que dispone de un entorno visual

para el desarrollo de los agentes y poder controlar el comportamiento de los mismos, además, pueden observarse las acciones, objetivos y las pre condiciones de los agentes para su correcto funcionamiento dentro del sistema.

5.1.8. Arquitectura de los agentes inteligentes

La arquitectura describe la manera en la que está construido un sistema, y la posibilidad que tiene este de descomponer en un sistema de software de componentes pequeños para determinar las relaciones internas de estos. Los agentes inteligentes pueden ser descompuestos para determinar las interacciones de éstos con el ambiente. [3] La arquitectura es un aspecto muy importante dentro del desarrollo de software, en este caso, es muy importante el definir una arquitectura que se adapte a las unidades de código desarrolladas previamente, según el enfoque del proyecto y teniendo en cuenta que la solución a entregar está basada en agentes inteligentes.

5.1.9. Arquitecturas deliberativas

En esta arquitectura el comportamiento y el conocimiento de los agentes se encuentran representados de una manera explícita a través de modelos simbólicos, compuestos de símbolos para referenciar entidades y estructuras, además, la toma de decisiones de los agentes desarrollados bajo esta arquitectura está basada en el razonamiento lógico o psicológico, partiendo de un estado inicial hacia un conjunto de planes que satisfacen los objetivos que el agente se plantee con respecto a su interacción con el ambiente, este comportamiento se denomina como BDI. [3]

5.1.10. Arquitecturas reactivas

Las arquitecturas reactivas tienen como característica principal la carencia de un modelo basado en símbolos y un razonamiento complejo. Las decisiones tomadas por los agentes en esta arquitectura se basan en la lectura del medio ambiente, el análisis del mismo y posteriormente la ejecución de una acción. [3] Esta arquitectura debido a su poca complejidad puede presentar ocasiones donde las problemáticas a las que se enfrenta el agente sean no solucionables.

5.1.11. Arquitecturas híbridas

Este tipo de arquitectura, integran aspectos de las arquitecturas deliberativa y arquitectura reactiva, donde el desarrollo del agente se da por subsistemas, siendo el deliberativo donde se almacene toda lógica representada en sistemas de símbolos y el reactivo donde se almacene la lógica carente de razonamiento complejo.

El desarrollo de agentes bajo esta arquitectura puede aplicarse en un modelo de capas, donde la interacción con el entorno puede darse por una o más capas, y transferir información de manera óptima entre ellas, brindando mejores acciones de intervención frente a los estímulos recibidos; dicha estructuración por capas puede darse de dos maneras, de forma vertical, donde una sola capa tiene acceso a los sensores y a los actuadores o de forma horizontal donde todas las capas tienen acceso a los sensores y actuadores. [3]

La estructuración de los agentes que implementan la arquitectura híbrida consta de las siguientes capas:

5.1.11.1. Capa reactiva

También se conoce como la capa central, es el núcleo del agente, donde se da lugar a la toma de decisiones acerca de las acciones a realizar sobre los estímulos recibidos del entorno.

5.1.11.2. Capa de conocimiento

Es la capa media de la arquitectura, dando lugar al conocimiento que adquiere el agente del medio, dicho conocimiento es adquirido de manera simbólica del entorno.

5.1.11.3. Capa social

Es la capa superficial del agente, donde se da lugar a la interacción con otros agentes, la interacción directa con el entorno, así como también deseos e intenciones.

5.1.12. Sistemas multiagente

El desarrollo de los agentes no siempre se da de forma individual, existen casos donde, es necesario desarrollar diversos agentes con roles definidos para facilitar el funcionamiento de éstos, así como el poder optimizar su trabajo, esto se denomina como sistema multiagente y el funcionamiento no depende solamente de un agente, sino varios que interactúan entre sí, con el fin de comunicarse, consultar información relevante entre ellos y establecer su rol dentro del sistema. [3]

Los sistemas multiagente son diseñados teniendo en cuenta una estructura que permita una operación efectiva, además de que puedan interactuar de forma productiva entre ellos, dicha estructura debe contemplar todos los aspectos relacionados con los procesos de comunicación entre agentes, refiriéndose al lenguaje que emplean los agentes para el envío y recepción de los mensajes, además de permitir el intercambio de información a un alto nivel.[3]

La capacidad de comunicación es un elemento vital para la percepción de los agentes, ya que es en este momento donde se da la interpretación y las acciones a ejecutar, esto conlleva a la creación de grupos de agentes llamadas sociedades en donde se establecen objetivos grupales además de los objetivos individuales, para este caso un agente individual puede entablar comunicaciones e intercambiar información con los demás agentes, con el fin de cumplir sus objetivos.

5.1.13. Comunicación entre agentes

Para que exista comunicación entre dos o más agentes es necesario que se comprendan en los aspectos sintácticos y semánticos de los posibles mensajes que se intercambien.

El proceso de comunicación entre agentes puede asimilarse con la comunicación humana, por una parte estaría el mensaje, lo que se quiere transmitir, el sentido que se le quiere dar y finalmente la acción que realiza el receptor cuando recibe el mensaje.

Existen varias maneras en que un agente puede indicar el sentido de su mensaje, esto puede ser dado por un requerimiento, una declaración o una pregunta. Los lenguajes de alto nivel con los que se comunican los agentes están constituidos básicamente por expresiones que establecen los actos comunicativos que un determinado agente puede realizar.

Existen diferentes métodos y formas de estandarizar la sintaxis y la semántica del lenguaje de comunicación para los agentes, entre ellas está KQML que consiste en un protocolo que ayuda a los agentes en la coordinación de las interacciones, y en el desarrollo e intercambio de conocimiento. Estas herramientas facilitan la labor de darle sentido a las comunicaciones entre agentes, pero en ocasiones suele no ser suficiente y es necesaria la implementación de lenguajes más complejos y de mayor compatibilidad entre los agentes interactivos.

5.1.14. ABLE

El entorno de construcción y aprendizaje de agentes ABLE es un frameworks, librería de componentes y kit de herramientas productivo, desarrollado en java con el objetivo de desarrollar agentes inteligentes haciendo uso del razonamiento y aprendizaje de las máquinas. Esta herramienta contiene componentes que proveen interfaces java basadas en clases que se emplean para la construcción de librerías en java beans llamados Able Beans, estos beans son incluidos por la librería y se usan para la lectura y escritura de texto e información de bases de datos.[4]

Las técnicas empleadas para el aprendizaje de las máquinas se basan en redes neuronales y árboles de decisiones, al ser una herramienta basada en clases, interfaces y java beans permiten que cualquier aplicación java tenga compatibilidad con este framework.

Las reglas de ABLE se estructuran con condiciones if-else, manteniendo la semántica de java, estas pueden procesar diversas reglas brindadas por el kit de herramientas, aunque también pueden ser creadas desde el cero. [4] Estas reglas pueden usarse para implementar reglas simples de negocio, y la razón para implementar este tipo de reglas es la externalización de la lógica de negocio de la aplicación, de este modo puede ser modificada sin aplicar ningún cambio en la aplicación; estas reglas pueden cambiar de forma dinámica y efectiva afectando la lógica de negocio, pero con la ventaja de no tener que realizar ningún tipo de programación ni de compilación.

5.1.15. JADE

ABLE contiene modelos que proveen reglas predefinidas, que pueden ser usadas con el fin de crear interfaces de interacción para usuario que deseen modificar las reglas. El framework de desarrollo de agentes en java (JADE) es un software que hace una implementación completa del lenguaje Java, se enfoca en simplificar la implementación de sistemas multiagente por medio de un software intermedio que cumple las especificaciones FIPA a través de un conjunto de herramientas gráficas que permitan el análisis del código de una manera profunda y de las fases de despliegue. [4] Un sistema basado en jade puede ser desplegado en varias máquinas sin importar que no tengan el mismo sistema operativo y la configuración puede ser cambiada de forma remota a través de una interfaz gráfica de usuario.

Otra de sus principales ventajas es que puede realizarse cambios en el tiempo de ejecución, pueden moverse agentes de una máquina a otra en el momento que se requiera, además cuenta con bajos requerimientos de hardware y no es necesario tener una versión de los controladores de java actualizada. [4]

Esta herramienta provee una ejecución de tareas simple pero potente, modelos de composición de agente por agente que están basados en el paradigma de intercambio de mensajes asíncronos. También se provee un servicio de páginas amarillas que soportan mecanismos de descubrimiento y muchas otras características avanzadas que facilitan el desarrollo e implementación de sistemas distribuidos.

5.1.16. Metodologías de desarrollo

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un

proyecto de desarrollo. Es un proceso de software detallado y completo. 2 Describen las actividades, procedimientos, técnicas, herramientas y documentos, en su conjunto, normados y comprendidos en un marco de trabajo. Sirven de soporte en la estructuración, planificación y control requeridos para lograr la conversión de una necesidad o un grupo de necesidades a un sistema de información de manera eficiente. 1

Las metodologías se basan en una combinación de los modelos de proceso genéricos. Definen definición de etapas, ingresos y salidas, restricciones, comunicaciones, tareas ordenadas y distribución de recursos, artefactos, roles y actividades, junto con prácticas y técnicas recomendadas. La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado.

5.1.17. Selección de la metodología de desarrollo de software

La siguiente tabla plantea las principales características de las metodologías de desarrollo de software tradicional y ágil, la misma fue elaborada a partir de la información presentada en los documentos, [5], [6], [7].

5.1.18. Selección de la metodología de desarrollo ágil

La siguiente tabla plantea las principales características de las metodologías de desarrollo de software ágiles, la misma fue elaborada a partir de la información presentada en el documento, [8].

5.1.19. Metodología de desarrollo SCRUM

Scrum es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo son aquellos proyectos de entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum es un modelo de desarrollo ágil caracterizado por:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados.

Según [9] los componentes de la metodología scrum se pueden describir de la siguiente manera:

Tradicionales	Ágiles
Proceso mucho más controlado, con numerosas políticas/normas	Proceso menos controlado, con pocos principios
Cierta resistencia a los cambios	Especialmente preparados para cambios durante el proyecto
Existe un contrato prefijado	No existe contrato tradicional o al menos es bastante flexible
Roles específicos	Roles flexibles
Más roles	Pocos roles
La arquitectura del software es esencial y se expresa mediante modelos	Menos énfasis en la arquitectura del software
Orientada a proyectos de cualquier tamaño	Orientada a proyectos pequeños
Equipos grandes y dispersos	Equipos pequeños
Proyectos de media-alta duración	Proyectos de poca duración
Proyecto cerrado	Proyecto abierto a cambios
Documentación rigurosa	Poca documentación
Alto coste de prototipado	Bajo coste de prototipado
Existe una planificación a seguir	Cambios sobre la marcha
Pruebas tardías y pesadas	Pruebas continuas
Burocrático y formal	Flexible y participativo

Tabla 5.1: Metodologías de desarrollo de software

1. Acumulación (backlog) de productos, en donde se deriva una lista a partir de las especificaciones de los productos.
2. Acumulación de corrida (sprint), una lista que cambia en forma dinámica sobre las tareas que se van a completar en la siguiente corrida.
3. Corrida, un periodo de 30 días en donde el equipo de desarrollo transforma la acumulación en software que se puede demostrar.
4. Scrum diaria, una reunión breve en donde la comunicación es la regla número uno. Los miembros del equipo necesitan explicar lo que hicieron desde la última reunión, si se toparon con obstáculos y lo que planean hacer antes de la siguiente scrum diaria.
5. Demo, software funcional que se puede demostrar al cliente.

En Scrum se realizan entregas parciales y regulares del producto final, centra su atención en las actividades de control y no especifica prácticas de Ingeniería [10], Scrum

Características	XP	Kendall y Kendall	DSDM	FDD	ASD
Enfoque	Iterativo, incremental	Iterativo, incremental	Iterativo	Iterativo	Iterativo, incremental
Periodo del ciclo de iteración	1 – 6 semanas	2 – 4 semanas	El 20% del tiempo total 80% de producto.	2 días - 2 semanas	4 – 8 semanas
Tamaño adecuado del proyecto y complejidad	Proyecto pequeño y sencillo	Para proyectos grandes y complejos	Todo tipo de proyectos	Proyectos a gran escala	Proyectos más pequeños y sencillos
Involucramiento del usuario	Activamente involucrado	A través del propietario del producto	A través de lanzamientos frecuentes	A través de informes	A través de lanzamientos frecuentes
Documentación	básica	básica	Más que XP y SCRUM	Más alto entre todos	básica
Principales prácticas	Simplicidad, Programación en pares	Marco de trabajo de SCRUM	Time boxing, MoSCoW, Prototipado	Modelado de objetos, desarrollo por característica, uso del diagrama UML	Time boxing, Risk Driven, basado en características
Desarrollo de características concurrentes	Posible	Posible	Posible	Posible	Posible

Tabla 5.2: Metodologías de desarrollo de software ágiles

parte de la base de que los procesos definidos funcionan bien sólo si las entradas están perfectamente definidas y el ruido, ambigüedad o cambio es muy pequeño. Por lo tanto, resulta ideal para proyectos con requerimientos inestables, ya que fomenta el surgimiento de los mismos.

5.1.20. Metodología de desarrollo Kendall y Kendall

Según esta metodología el ciclo de vida de un Sistema creada por estos dos autores consta de siete partes: Cada fase se explica por separado, pero nunca se realizan como pasos aislados, es posible que algunas actividades se realicen de manera simultánea, y algunas de ellas podrían repetirse.

A continuación, se listan las siete etapas de la metodología Kendall y Kendall tomadas de [9].

1. Identificación de problemas, oportunidades y objetivos: en esta etapa se deberá descubrir lo que la organización intenta realizar, luego determinar si el uso de los sistemas de información apoyaría a la organización para alcanzar sus metas
 2. Determinación de los requerimientos de información: Esto se hace a partir de los usuarios particularmente involucrados, para determinar los requerimientos de información dentro de una organización pueden utilizarse diversos instrumentos, los cuales incluyen: muestreo, el estudio de los datos y formas usadas para la organización, la entrevista, los cuestionarios; la observación de la conducta de quien tomó las decisiones.
 3. Análisis de las necesidades del sistema: Se analizan las necesidades propias del sistema. También se analizan las decisiones estructuradas por realizar, que son decisiones donde las condiciones, condiciones alternativas, acciones y reglas de acción podrán determinarse.
 4. Diseño del sistema recomendado: Se usa la información recolectada con anterioridad y se elabora el diseño lógico de sistemas de información, esta etapa también incluye el diseño de los archivos o la base de datos que almacenará aquellos datos requeridos por quien toma las decisiones en la organización.
 5. Desarrollo y documentación del software: Dentro de las técnicas estructuradas para el diseño y documentación del software se tienen: el método HIPO, los diagramas de flujo, los diagramas Nassi-Schneiderman, los diagramas Warnier-Orr y el pseudocódigo es aquí donde se transmite al programador los requerimientos de programación.
 6. Pruebas y mantenimiento del sistema: Todo sistema de información debe probarse antes de ser utilizado, ya que el costo es menor si se detectan los problemas antes de que entre en funcionamiento.
 7. Implantación y evaluación del sistema: Esta es la última etapa del desarrollo del sistema, esto incluye el adiestramiento que el usuario requerirá. Uno de los criterios fundamentales que debe satisfacerse, es que el futuro usuario utilice el sistema desarrollado.
-

5.1.21. Medidas de bioseguridad

A continuación se presenta una lista con las medidas de protección básicas contra el nuevo coronavirus, las mismas que han sido obtenidas del Protocolo de prevención de COVID-19 [11] presentado por parte del ministerio de salud pública de la república del Ecuador, y también de las entrevistas realizadas a los diferentes centros de salud, que constan en el Informe de entrevista (Anexo 2).

- Lavarse las manos con frecuencia con agua y jabón.
 - Cubrirse al toser o estornudar.
 - No escupir.
 - Distanciamiento social.
 - Evite tocarse los ojos.
 - Uso de mascarilla.
 - Uso de guantes.
 - Al momento de acudir a un centro de salud, hacerlo de preferencia solo o con un acompañante.
-

6. Materiales y métodos

En esta sección, constan los procesos llevados a cabo en el desarrollo del presente Trabajo de Titulación (TT), métodos científicos, técnicas empleadas y la metodología de desarrollo de software empleada para el cumplimiento de los objetivos, los recursos que se emplearon y los participantes que interactuaron en el TT.

6.1. Contexto

El presente Trabajo de Titulación (TT) se desarrolló en la Universidad Nacional de Loja, en la Facultad de Energía las Industrias y los Recursos Naturales No Renovables (FEIRNNR), en la Carrera de Ingeniería en Sistemas (CIS); el TT está enfocado en la implementación de un agente inteligente a un sistema de llamadas telefónicas, el cual se encuentra dividido en tres fases, mismas que se detallan en el siguiente apartado.

6.2. Proceso

Etapas 1: Proceso de ingeniería de requisitos

- Realizar un estudio de documentación sobre trabajos similares al problema estudiado para obtener requerimientos.
- Análisis de los requerimientos obtenidos en el estudio de documentación.
- Especificar los requerimientos utilizando el estándar IEEE 830.
- Validar los requisitos mediante la técnica de la revisión y prototipado.

Etapas 2: Desarrollo del módulo de software

- Modelar el diseño arquitectónico en base a los requerimientos de una aplicación web.
- Diseñar el módulo de software mediante Diagramas de Clases y Paquetes utilizando notación UML.
- Desarrollo del agente inteligente en Netbeans utilizando la metodología XP.
- Generar las pruebas pertinentes del software según la metodología XP.

Etapas 3: Evaluación del diseño de software

- Generar pruebas de funcionalidad y usabilidad.
- Analizar los resultados obtenidos en las pruebas de funcionalidad y usabilidad.

6.3. Recursos

6.3.1. Recursos Científicos

6.3.1.1. Métodos

- **Método Analítico**
Se hace uso del método analítico para dividir el trabajo realizado en diferentes etapas y actividades a seguir durante la realización del mismo, ayudando así a la realización de los diferentes objetivos específicos. [12]
- **Método Científico**
Obtenido de [13], se toman las etapas del método científico para el desarrollo del TT: Definición y planteamiento del problema, formulación de la hipótesis la cual se encuentra dentro de la pregunta de investigación “¿El uso de agentes inteligentes puede agilizar el proceso de atención mediante llamadas telefónicas en un centro de salud, considerando la saturación de los centros de salud debido al estado de emergencia sanitaria?”, recolección, procesamiento y análisis de datos los cuales se encuentran en la sección de Resultados, se da la contraste de los datos con la hipótesis (ver sección Discusión), y por último se elabora las conclusiones (ver sección Conclusiones).
- **Método inductivo-deductivo**
Utilizado en la recolección de datos, análisis de la información e interpretación de los hechos y descubrimiento de nuevos procedimientos. El método de razonamiento inductivo se utilizó para poder obtener la secuencia lógica que debe tener el comportamiento del sistema desarrollado (Revisar Anexo x) en base a las respuestas dadas por las personas entrevistadas (Revisar Anexo x). El método de razonamiento deductivo ha sido utilizado para poder definir qué tareas dentro del desarrollo del sistema se deben considerar de importancia (Revisar Anexo x), así como también para poder obtener la lista de funcionalidades que satisfacen los requerimientos de los entrevistados (Revisar Anexo x).

6.3.1.2. Técnicas

- **Estudio de Caso**
En este método se utilizó para la revisión sistemática de la literatura relacionada al tema del trabajo planteado, ayudando así a aclarar el estado del arte, identificado
-

trabajos relacionados y ayudando a mejorar el enfoque del proyecto. (Revisar Anexo 1: RSL)

- Muestreo no probabilístico por conveniencia

Para la aplicación de las entrevistas a los diferentes centros de salud en la ciudad de Loja, ya que el mismo permite seleccionar aquellos casos accesibles que acepten ser incluidos. Esto, fundamentado en la conveniente accesibilidad y proximidad de los sujetos para el investigador. [14]

- Entrevistas

Para recaudar la información que ayude a la obtención de la lista final de requisitos del presente proyecto, se realizaron entrevistas en distintas instituciones de salud pública y privada como hospitales y clínicas de la ciudad de Loja. Las evidencias de dicha información se las encuentra en el anexo 2 (Revisar anexo 2: Informe de entrevista) y seguido del informe de elicitación de requerimientos (Revisar anexo 3: Elicitación de Requerimientos).

- Encuestas

Dirigidas a un primer grupo en el campo de la medicina, en donde constan estudiantes y profesionales de la salud, para poder obtener una lista de síntomas, los mismos que fueron usados en la aplicación para tener una clasificación de los pacientes más precisa y poder tomar las respectivas decisiones.

En el segundo grupo las encuestas estuvieron dirigidas a los estudiantes del 10mo ciclo de la carrera de ingeniería en sistemas del Área de la Energía las Industrias y los Recursos No Renovables, con la finalidad de validar la aplicación y analizar la funcionabilidad de la misma.

El tercer grupo de encuestados fue definido de forma aleatoria para poder recibir opiniones y sugerencias desde un punto de vista no técnico, dándonos un punto de vista como usuario final de la aplicación.

//Conclusión —lo cual fue de gran importancia ya que se pudo recolectar diferentes opiniones y sugerencias de parte de los encuestados para de esta manera corregir errores o hacer nuevas mejoras a la aplicación.

- Prototipado

Para modelar el software final, permitiendo así obtener información objetiva del producto y efectuar las diferentes pruebas en la aplicación a sus diferentes atributos, permitiéndonos implementar mejoras sin necesidad de que el producto final esté disponible.

- Técnica de experimentación

Posiblemente se use al probar el software

6.3.2. Recursos Académicos

- Guías de desarrollo
- Foros
- Podcast
- Artículos
- Documentación gris
- Bases de datos científicas
- Revistas
- Conferencias
- Estándares

6.3.3. Recursos Técnicos

- Editor de LaTeX, Overleaf
- Metodología de Burmeister
- Estándar IEEE 830
- Framework Netbeans
- Lenguaje de programación JAVA
- Librerías de desarrollo para JAVA

6.3.4. Recursos Éticos

- Consentimiento informado
 - Políticas de Privacidad
 - Estándares
-

6.4. Participantes

- Sandro Michael Córdova Carrión como investigador del Trabajo de Titulación (TT).
 - Ing. José Oswaldo Guamán Quinche, Mg.Sc. como director del TT, orientando de manera académica y técnica el desarrollo de los objetivos planteados para el presente TT.
 - "Nombre" como profesional experto en el área de salud y actualmente "Ocupación".
 - "Nombre" como asistente personal de llamadas al centro de salud "Institución".
 - "Nombre" como encargado del departamento de comunicaciones en el centro de salud "institución".
-

7. Resultados

7.1. Análisis de metodologías para el desarrollo de software

La implementación de una metodología tradicional conlleva a un gran estudio previo al inicio del proyecto, es necesario el definir un proceso claro y muy bien planificado para poder cumplir con los tiempos y presupuestos estimados. Al ser una metodología encaminada a proyectos grandes requiere de un mayor tiempo para su desarrollo, más personal para poder definir roles y mayor costo de inversión.

Por otro lado, se encuentran las metodologías de desarrollo ágiles, diseñados para proyectos pequeños, con menor planificación, menor numero de roles, mayor adaptabilidad al cambio y un coste de inversión menor, dejándola así como la metodología idónea para el proyecto.

Dentro del análisis de las metodologías para el desarrollo de software ágiles se determinó que la metodología Kendall y Kendall es la idonea para poder llevar a cabo este proyecto, debido a las características mencionadas de las metodologías ágiles, a esto se le suma que los autores de dicha metodología consideran que fue desarrollada para ser aplicada en cualquier tipo de sistema de información.

7.2. Entrevista para la elicitación de requisitos

El cumplimiento de esta actividad se dio gracias a la entrevista como técnica de elicitación de requisitos, el resultado obtenido para esta actividad es el informe final de la entrevista el cual contiene a detalle, los datos de las personas entrevistadas, las preguntas aplicadas, las respuestas por parte del personal, el respectivo análisis de cada una de estas respuestas y por último la lista preliminar de los requerimientos obtenidos, los mismos que se encuentran listados en la tabla 7.1.

7.3. Historias de usuario

Las historias de usuario obtenidas se pueden ver en la tabla 7.2.

Código	Requisitos
01	Comunicación con los médicos.
02	Comunicación los diferentes departamentos.
03	Informar costos.
04	Apartado para pacientes con COVID-19.
05	Informar sobre medidas de bioseguridad.
06	Recetar en caso de emergencia.
07	Tiempo de iteración de 6min.
08	Voz aguda y agradable.
09	Gestión de módulos.
10	Gestión de información general.
11	Gestión de citas.
12	Gestión de médicos.

Tabla 7.1: Requisitos preliminares.

7.4. Requisitos funcionales

Los requisitos funcionales obtenidos se pueden ver en la tabla 7.3.

7.5. Requisitos no funcionales

Los requisitos no funcionales se pueden ver en la tabla 7.4.

7.6. Metodología para el desarrollo de software

En la primera fase de la metodología se efectuó el análisis del estado actual de cómo se lleva el proceso de ingreso de una llamada al centro de salud y posteriormente determinar los problemas existentes, se hizo uso de la entrevista como método para recaudar la información que ayude a la obtención de la lista final de requisitos del presente proyecto, se realizaron entrevistas en distintas instituciones de salud pública y privada como hospitales y clínicas de la ciudad de Loja. El proceso para la identificación

Código	Nombre de la historia
01	Transferir llamada.
02	Generar cita médica.
03	Clasificación del paciente.
04	Medidas de bioseguridad.
05	Recetar en casos de emergencia.
06	Crear módulo.
07	Visualizar módulos.
08	Actualizar módulos.
09	Borrar módulos.
10	Agregar cita.
11	Visualizar cita.
12	Actualizar cita.
13	Borrar cita .

Tabla 7.2: Historias de usuario.

de problemas, oportunidades y objetivos, fue desarrollada y se detalla en el anteproyecto que respalda a este trabajo de titulación.

Para la fase dos, se hizo uso de la entrevista como técnica para la elicitación de los requisitos, aquí se realizó el análisis de las principales actividades realizadas por los funcionarios de las diferentes instituciones al momento de interactuar con el paciente que se comunica con el centro de salud y se determinó la siguiente lista preliminar de requisitos que resume todas estas actividades involucradas.

Para el desarrollo de la fase tres, se realizó la determinación de los requerimientos a través de la especificación de los requisitos que plantea el estándar IEEE830, gracias a esto se pudo determinar la lista de historias de usuario, requisitos funcionales, requisitos no funcionales, los mismos que se encuentran detallados en revisar anexo especificación de requisitos IEEE830.

Las fases cuatro y cinco de la metodología corresponde al diseño del sistema recomendado, para el desarrollo de esta fase se hizo uso del esquema planteado por SCRUM el cual ayudó a cumplir con los entregables requeridos en los lapsos de entrega determinados, se hizo uso de las iteraciones en donde cada iteración tiene la estructura de: planificación, diseño, codificación, pruebas y resultados. El resultado de esta fase se puede evidenciar en el documento de desarrollo y documentación del software, detallado en el .

La fase 6 y 7 corresponden a la aplicación y evaluación del software. Estado: En proceso

Código	Requisitos
01	Comunicación con los médicos.
02	Comunicación los diferentes departamentos.
03	Informar costos.
04	Apartado para pacientes con COVID-19.
05	Informar sobre medidas de bioseguridad.
06	Recetar en caso de emergencia.
07	Tiempo de iteración de 6min.
08	Voz aguda y agradable.
09	Gestión de módulos.
10	Gestión de información general.
11	Gestión de citas.
12	Gestión de médicos.

Tabla 7.3: Requisitos funcionales.

Código	Requisitos
01	Rendimiento.
02	Seguridad.
03	Fiabilidad.
04	Disponibilidad.
05	Mantenibilidad.
06	Portabilidad.
07	Usabilidad.

Tabla 7.4: Requisitos no funcionales.

8. Conclusiones

9. Recomendaciones

Bibliografía

- [1] B. López, *Inteligencia Artificial*. Reading: Nuevo Laredo, Tam, 2005.
- [2] J. Sanz, “Metodologías para el desarrollo de sistemas multi-agente. (German) [On the electrodynamics of moving bodies],” *Annalen der Physik*, vol. 7, no. 10, pp. 51–63, 2003.
- [3] A. Samarin, *AGENTES SOFTWARE Y REDES ACTIVAS*. 2017.
- [4] A. Madruga, “Agentes inteligentes: Tipos de plataformas. (German) [On the electrodynamics of moving bodies],” 2008.
- [5] M. D. Desarrollo De Software and E. Gabriel Pacienza, “FACULTAD DE QUÍMICA E INGENIERIA ”FRAY ROGELIO BACON” PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA SANTA MARIA DE LOS BUENOS AIRES Cátedra Seminario de Sistemas,” tech. rep.
- [6] F. González, S. Calero, and D. Loaiza, “Comparación de las metodologías cascada y ágil para el aumento de la productividad en el desarrollo de software ,” *Universidad Santiago de Cali, Facultad de Ingeniería, Programa de Ingeniería Industrial*, vol. 11, 2019.
- [7] J. G. Navarro, “Estudio Comparativo De Metodologías, Herramientas Y Wiki De Soporte Para La Gestión De Proyectos De Desarrollo De Software,” *Universitat Oberta de Catalunya*, p. 90, 2018.
- [8] J. M. Q. Xavier Quiñónez-Ku, Juan Casierra Cavada, Luis Herrera-Izquierdo, “Análisis Comparativo de metodologías Ágiles de desarrollo de software : una revisión bibliográfica Comparative analysis of agile software development methodologies : a bibliographic review,” *ResearchGate*, no. June, 2019.
- [9] K. Kendall and J. Kendall, *Analisis Y Diseño De Sistemas*. 2011.
- [10] A. Peralta, “Metodologia scrum,” *Universidad ORT Uruguay*, p. 12, 2003.
- [11] A. A. Hidalgo, “Lineamientos para la prevención COVID-19 e Inocuidad de los Alimentos REGISTRO DE REVISIÓN Y APROBACIÓN DEL DOCUMENTO Acción Nombre / Cargo Institución Firma y Fecha Elaborado por,” p. 23, 2020.

-
- [12] A. Rodríguez Jiménez and A. Omar Pérez Jacinto, “Métodos científicos de indagación y de construcción del conocimiento,” pp. 179–200, 2017.
 - [13] R. Ruiz, “El Metodo Cientifico y sus Etapas,” *Mexico*, vol. 2, p. 79, 2011.
 - [14] T. Otzen and C. Manterola, “Técnicas de Muestreo sobre una Población a Estudio Sampling Techniques on a Population Study,” Tech. Rep. 1, 2017.
-

A. Anexo I



UNIVERSIDAD NACIONAL DE LOJA



**FACULTAD DE ENERGIA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

**REVISIÓN SISTEMÁTICA DE LITERATURA PARA UN SISTEMA DE
LLAMADAS TELEFÓNICAS PARA LA PRESTACIÓN DE SERVICIOS
MEDIANTE UN AGENTE INTELIGENTE**

Nombre:

- Sandro Michael Córdova Carrión

Docente: Ing. Pablo Ordoñez

Ciclo: Noveno Ciclo

Loja-Ecuador

Introducción

El presente proyecto de trabajo de titulación pretende establecer un servicio de información y de asistencia para casos de COVID-19 mediante la automatización de llamadas al establecer un agente inteligente de asistencia, con este servicio los usuarios tendrán atención las veinticuatro horas, en donde los mismos podrán obtener desde información vital en caso de emergencia hasta información superficial despejando dudas en cuanto a síntomas presentados.

El proyecto se enfoca principalmente en aquellos grupos de personas que no pueden hacer uso de aplicaciones web o móviles debido a su poca relación con la tecnología como adultos mayores o personas con alguna incapacidad, con esto, el proyecto aportaría a la no propagación del virus al funcionar como fuente confiable de información para aquellas personas que no pueden hacer uso de otro servicio virtual.

Objetivos

Objetivo General

- Recapitular la información existente respecto a trabajos relacionados con un sistema de llamadas telefónicas para la prestación de servicios mediante un agente inteligente.

Objetivo Específico

- Realizar una revisión exploratoria de trabajos relacionados a un sistema de llamadas telefónicas para la prestación de servicios mediante un agente inteligente.
- Analizar los trabajos relacionados existentes y generar una tabla con las características principales de dichos trabajos.
- Determinar las aportaciones del tema de investigación a otras investigaciones con el mismo enfoque.

Preguntas de investigación

¿Existe un sistema de llamadas gestionado por agentes inteligentes?

¿Qué clase de servicios prestan los sistemas de llamadas gestionados por agentes inteligentes?

Especificar las preguntas de investigación a través del acrónimo PICOC le ayuda a crear una cadena de búsqueda completa:

- **Población:** Universidad
- **Intervención:** Sistemas de llamadas que usen agentes inteligentes
- **Comparación:** Sistemas de llamadas tradicional
- **Salida:** Guía para la implementación de un agente inteligente a un sistema de llamadas
- **Contexto:** Academia, COVID-19

Palabras clave:

- Automatización de llamadas.
- Agente inteligente.
- Asistencia telefónica.
- Software.
- Servicios digitales.



Cadenas iniciales

(((((("All Metadata": Automatización de llamadas.) AND "All Metadata": Agente inteligente) AND "All Metadata": Asistencia telefónica) AND "All Metadata": Software))

(((((("All Metadata": Call automation.) AND "All Metadata": Intelligent agent) AND "All Metadata": Telephone support) AND "All Metadata": Software)

- Cero resultados.
- Eliminar “Call automation”.

Primera versión

((("All Metadata": Intelligent agent) AND "All Metadata": Telephone support) AND "All Metadata": Software)

Correcciones:

- Siete resultados.
- Eliminar “Telephone support”, “software”.
- Agregar “Multimedia systems”, “telephony”.

Segunda versión

((("All Metadata": Intelligent agent) AND "All Metadata": Multimedia systems) AND "All Metadata": telephony)

Correcciones:

- Eliminar “multimedia systems”.
- Agregar “mobile phone”.

Tercera versión

((("All Metadata": Intelligent agent) AND "All Metadata": mobile phone) AND "All Metadata": telephony)

Cuarta versión

Correcciones:

- Agregar “conversational agent”.
- Eliminar “mobile phone”.
- Agregar “dialogue agent”.
- Agregar “agente interactivo”

IEEE

(((((("Todos los metadatos": soporte telefónico) O "Todos los metadatos": teléfono) Y "Todos los metadatos": agente inteligente) O "Todos los metadatos": agente conversacional) Y "Todos los metadatos": diálogo del agente)

(((((("All Metadata":phone support) OR "All Metadata":telephone) AND "All Metadata":intelligent agent) OR "All Metadata":agent conversational) AND "All Metadata":agent dialogue)

[https://ieeexplore.ieee.org/search/searchresult.jsp?action=search&newsearch=true&matchBoolean=true&queryText=\(\(\(\(\(%22All%20Metadata%22:phone%20support\)%20OR%20%22All%20Metadata%22:telephone%20\)%20AND%20%22All%20Metadata%22:intelligent%20agent\)%20OR%20%22All%20Metadata%22:agent%20conversational%20\)%20AND%20%22All%20Metadata%22:agent%20dialogue\)&ranges=2015_2021_Year](https://ieeexplore.ieee.org/search/searchresult.jsp?action=search&newsearch=true&matchBoolean=true&queryText=(((((%22All%20Metadata%22:phone%20support)%20OR%20%22All%20Metadata%22:telephone%20)%20AND%20%22All%20Metadata%22:intelligent%20agent)%20OR%20%22All%20Metadata%22:agent%20conversational%20)%20AND%20%22All%20Metadata%22:agent%20dialogue)&ranges=2015_2021_Year)

ACM

[Full Text: intelligent agent] AND [Abstract: telephone support] AND [Full Text: telephony] AND [Full Text: dialogue agent] AND [Full Text: conversational agent] AND [Publication Date: (01/01/2015 TO 01/31/2021)]

<https://dl.acm.org/action/doSearch?fillQuickSearch=false&expand=dl&AfterMonth=1&AfterYear=2015&BeforeMonth=1&BeforeYear=2021&AllField=Fulltext%3A%28Intelligent+agent%29+AND+Abstract%3A%28Telephone+support%29+AND+Fulltext%3A%28telephony%29+AND+Fulltext%3A%28dialogue+agent%29+AND+Fulltext%3A%28conversational+agent%29&pageSize=20&startPage=1>

SCOPUS

(TITLE-ABS-KEY (phone AND support) OR TITLE-ABS-KEY (telephone) AND TITLE-ABS-KEY (intelligent AND agent) OR TITLE-ABS-KEY (agent AND conversational) OR TITLE-ABS-KEY (agent AND dialogue)) AND PUBYEAR > 2014

[https://www.scopus.com/results/results.uri?src=s&st1=&st2=&sot=b&sdt=b&origin=searchbasic&sl=170&s=\(TITLE-ABS-KEY%20\(phone%20support\)%20OR%20TITLE-ABS-KEY%20\(telephone\)%20AND%20TITLE-ABS-KEY%20\(intelligent%20agent\)%20OR%20TITLE-ABS-KEY%20\(agent%20conversational\)%20OR%20TITLE-ABS-KEY%20\(agent%20dialogue\)\)&yearFrom=2015&yearTo=Present](https://www.scopus.com/results/results.uri?src=s&st1=&st2=&sot=b&sdt=b&origin=searchbasic&sl=170&s=(TITLE-ABS-KEY%20(phone%20support)%20OR%20TITLE-ABS-KEY%20(telephone)%20AND%20TITLE-ABS-KEY%20(intelligent%20agent)%20OR%20TITLE-ABS-KEY%20(agent%20conversational)%20OR%20TITLE-ABS-KEY%20(agent%20dialogue))&yearFrom=2015&yearTo=Present)

Fuentes de datos

Fuente	Resultados
ACM Digital Library	66
SCOPUS	52
Explore Digital Library	67

Criterios de inclusión

- Estudios que implementen un agente inteligente.
- Fecha de publicación: 2015-2021
- En el título, resumen y conclusiones, contiene al menos dos palabras de la cadena de búsqueda.
- Los estudios poseen información relevante para las preguntas de investigación
- Trabajos primarios.

Criterios de exclusión

- Estudio se aleja del campo de análisis.
- Idioma diferente a inglés o español.
- Acceso restringido al documento.
- El documento presenta información incompleta.
- Estudios duplicados.
- Documento fuera de contexto.



Metodología

La presente revisión sistemática de literatura se basa en la metodología de revisiones sistemáticas de Bárbara Kitchenha, para la revisión, selección y extracción de información haciendo uso de la siguiente estructura:

Planteamiento de preguntas de investigación, establecer palabras clave, definir el método de revisión, fuentes y estrategias de búsqueda, cadenas de búsqueda, criterios de selección y exclusión de estudios, extracción de información, estudios incluidos y excluidos.

Además, se hace uso de la aplicación web SESRA para respaldar el proceso de revisión sistemática de literatura, para almacenar y organizar los estudios.

Desarrollar e implementar

Desarrollar un software es crear un software, es como hacer un programa desde el principio, diseñarlo, etc.

Implementar un software es hacer uso de él.

Calidad

¿El estudio muestra información para el desarrollo de un agente inteligente?

¿El estudio muestra información para la implementación de un agente inteligente?

¿El estudio expone los beneficios de usar un agente inteligente en un sistema de llamadas telefónicas?

¿El estudio expone las ventajas de usar un agente inteligente?

Titulo	¿Cuáles son las ventajas de usar un agente inteligente?	¿Qué servicios ofrece el agente inteligente?	¿Cómo se da la implementación de un agente inteligente?	Tecnologías y/o herramientas
Application Prospect of Artificial Intelligence Technology Basted on the Large Call Center	<ul style="list-style-type: none"> - Aprendizaje de idiomas. - Aplicaciones educativas. - Permitir una experiencia de conversación más inmersiva. - Recibir comentarios específicos a los participantes. 	Agente de diálogo virtual de código abierto que puede servir como entrevistador de trabajo para solicitudes de capacitación laboral.	Se da en forma de aplicación web, permite el acceso a los dispositivos de entrada de audio y video de la computadora a través del navegador web.	Está escrita en HTML, CSS y Javascript. La API de captura y transmisión de medios5 permite el acceso a los dispositivos de entrada de audio y video de la computadora. Explora la creación de un avatar prototípico usando Blender1 y Unity3D2.
Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots	<ul style="list-style-type: none"> - Proporcionar una entrada de barrera baja para los usuarios, permitiéndoles interactuar de manera intuitiva con servicios, recursos y datos. - Sistemas de diálogo no es necesario aprender una interfaz. - Autoservicio al cliente y ayuda automatizada 	Asistente personal de dispositivos como: <ul style="list-style-type: none"> - Alexa - Google Meena - Facebook Blender bot 	n/a	Aprendizaje de refuerzo. La aplicación de POMDP como modelo estadístico para los sistemas de diálogo hablado. Lista de corpus: <ul style="list-style-type: none"> - Repositorio de Ruder - Serban et al.
Daily life support at home through a virtual support partner	Proporciona compañía y asistencia a los adultos mayores en dominios relacionados con la salud.	Asistente de actividades diarias y necesidades de seguridad de las personas mayores.	Se da en forma de aplicación móvil.	<ul style="list-style-type: none"> - Módulos de reconocimiento de usuarios. - Módulos de reconocimiento de voz. - Análisis del contexto ambiental. - Marco de gestión de diálogo multimodal VSP y Avatar. - Base de conocimientos. - Servicios de seguridad y actividades diarias en el hogar. - Módulos de conexión de sensores y sensores. - Reconocimiento de emociones.

Enabling Aid in Remote Care for Elderly People via Mobile Devices: The MobiCare Case Study	Abre un canal de comunicación y contribuye al cuidado de personas mayores desconectadas digitalmente.	Asistente tecnológico que habilita ayudas en el cuidado de personas mayores.	Aplicación para teléfonos inteligentes.	<ul style="list-style-type: none"> - Firebase Realtime fue adoptada para permitir el almacenamiento y sincronización de la información. - Diseño y usabilidad centrados en el usuario.
Extending a Conventional Chatbot Knowledge Base to External Knowledge Source and Introducing User Based Sessions for Diabetes Education	Permite a cada usuario interactuar con el sistema en sus propias palabras, en lugar de utilizar una de las pocas formas preestablecidas para interactuar.	Ayuda en general a miembros de la comunidad y pacientes con diabetes interactuando con ellos durante una sesión de preguntas y respuestas.	Chatbot basado en web (VDMS).	<ul style="list-style-type: none"> - Utiliza Wikipedia información, junto con su propio conocimiento local codificado a mano.
Hierarchical Recurrent Attention Networks for Context-Aware Education Chatbots	n/a	Diálogo educativo que ayuda a los estudiantes, en la Universidad de Lyon. Es de atención recurrente jerárquica (HRAN) para múltiples turnos generación de respuestas y respuesta a preguntas.	Arquitectura en red de red jerárquica.	<ul style="list-style-type: none"> - Métodos HRED y VHRED.
Interpreting Human Responses in Dialogue Systems using Fuzzy Semantic Similarity Measures	Se desarrolla a través de diálogo lingüístico para proporcionar un tipo de servicio o asesoramiento.	Sistema de diálogo para permitir que una máquina comprenda semánticamente expresiones humanas en un contexto dado.	n/a	<ul style="list-style-type: none"> - Similitud semántica FUSE - PatternScript scripting language
kBot: Knowledge-Enabled Personalized Chatbot for Asthma Self-Management	Monitorización continua del paciente, adherencia a la medicación y seguimiento de señales de salud relevantes y datos ambientales.	Se encarga de la recopilación, el procesamiento y la gestión del diálogo de los datos.	Aplicación de Android.	<ul style="list-style-type: none"> - Servidor en JSON7 - Los datos se almacenan en Elasticsearch 8, un motor de búsqueda de texto completo. - Base de datos NoSQL. - Procesamiento de diálogo, contexto y análisis de la intención del usuario: DialogFlow19

Learning to predict the adequacy of answers in chat-oriented humanagent dialogs	Crear un conjunto de datos de diálogos anotados y resultados exploratorios al especificar un conjunto de características para entrenar un clasificador.	El clasificador es capacitado en un conjunto de diálogos recopilados y anotados de origen colectivo entre usuarios y diferentes motores de chatbot.	n/a	Clasificadores de texto: <ul style="list-style-type: none"> - Etiqueta de mayoría “baseline”, Multinomial Naïve Bayes (MNB) - Regresión logística (LR) - Clasificación de vectores de soporte C (SVC) - Árbol de decisión.
Voice-Based Monitoring and Control System of Electronic Appliance Using Dialog Flow API Via Google Assistant	Se hace un análisis al asistente de google que mediante la API dialog flow se podría implementar en un proyecto	Procesar la conversación de los humanos basada en Inteligencia artificial.	<ul style="list-style-type: none"> - Aplicación para teléfonos inteligentes. - Aplicación web. 	<ul style="list-style-type: none"> - API Dialog Flow. - Asistente de google.
FormYak: Converting forms to conversations	<ul style="list-style-type: none"> - Solicita la información necesaria para completar documentos. - Disminuye el trabajo de los funcionarios al transcribir todos los formularios. - Genera automáticamente un documento o llena una base de datos con la información recopilada. 	Adapta el proceso de llenado de formas para adquirir información del cliente para la interacción basada en conversaciones.	<ul style="list-style-type: none"> - Aplicación para teléfonos inteligentes. - Aplicación web. 	<ul style="list-style-type: none"> - FormYak, que facilita la creación de conversaciones - Genera scripts de diálogo compatibles con FormFlow de Microsoft (utilizando NodeJS).
Collaborative Human-Machine Interaction in Mobile Phone Support Centers: A Case Study	<ul style="list-style-type: none"> - Responde preguntas. - Explicar procedimientos. - Ayuda con la resolución de problemas de diagnóstico. 	<ul style="list-style-type: none"> - Automatiza las operaciones del centro de llamadas. - Manteniendo un alto nivel de satisfacción del cliente. 	Agente de centro de llamadas	<ul style="list-style-type: none"> - Incorpora aspectos de la conversación humana. - Capturar detalles relevantes relacionados con la intención (para poder intervenir en la conversación). - Planificación colaborativa es

				más elaborado en el modelo SharedPlans.
Acceptability, Feasibility, And Preliminary Efficacy Of A Theorybased Relational Embodied Conversational Agent Mobile Phone Intervention To Promote Hiv Medication Adherence In Young Hivpositive African American Msm	Proporciona información, motivación y habilidades de comportamiento.	Intervención de agente conversacional encarnado similar a un avatar humano para abordar la adherencia al TAR.	Aplicación para teléfonos inteligentes.	n/a
Building an Intelligent Call Distributor	Combina procesamiento del lenguaje hablado y escrito.	Enruta las llamadas al agente más apropiado de acuerdo con las reglas de enrutamiento construidas por un clasificador de texto.	Agente de centro de llamadas.	<ul style="list-style-type: none"> - El Vector de soporte Machine y Naïve Bayes, para clasificar los comandos de texto, y luego elegir un método adecuado. - La máquina de vectores de soporte (SVM) desarrollada por Vapnik.

Conclusiones

Existen muy pocos agentes inteligentes de asistencia telefónica, la mayoría de estos agentes se intentan reemplazar con la ayuda de las extensiones telefónicas.

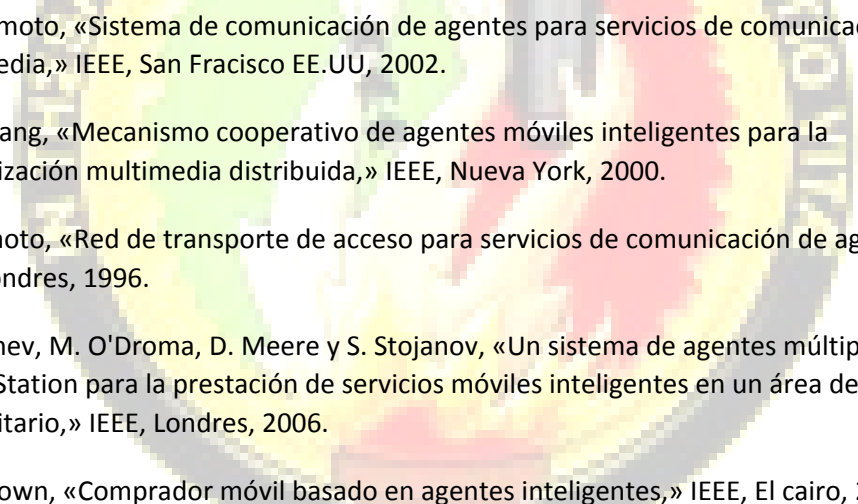
En su mayoría, los agentes inteligentes de asistencia son usados para mensajería, la contratación de servicios y dentro de aplicaciones web de empresas para brindar atención al cliente.

El uso de agentes inteligentes dentro de la telefonía por parte de algunas empresas es netamente lucrativo, con el fin de informar sobre la contratación de un determinado servicio y concretar ventas. Con este proyecto se plantea el uso de agentes inteligentes para informar a los usuarios y resolver dudas que pueden ser de vital importancia.

URL SESRA:

http://sesra.net/systematic_review/team/id/558

Bibliografía

- 
- [1] R. Kishimoto, «Sistema de comunicación de agentes para servicios de comunicación multimedia,» IEEE, San Francisco EE.UU, 2002.
 - [2] Y.-H. Wang, «Mecanismo cooperativo de agentes móviles inteligentes para la sincronización multimedia distribuida,» IEEE, Nueva York, 2000.
 - [3] R. Ishimoto, «Red de transporte de acceso para servicios de comunicación de agentes,» IEEE, Londres, 1996.
 - [4] I. Ganchev, M. O'Droma, D. Meere y S. Stojanov, «Un sistema de agentes múltiples basado en InfoStation para la prestación de servicios móviles inteligentes en un área de campus universitario,» IEEE, Londres, 2006.
 - [5] R. A. Brown, «Comprador móvil basado en agentes inteligentes,» IEEE, El Cairo, 2009.

file:///C:/Users/Usuaría/Downloads/113-Article%20Text-385-1-10-20161209.pdf

B. Anexo II

Especificación de requisitos de software IEEE 830

Proyecto: Diseño de un sistema de información sobre COVID-19 mediante llamadas telefónicas con la ayuda de un agente inteligente

Ficha de documento

Fecha	Revisión	Autor	Verificado y Validado
26/06/2021	0.0.1	Sandro Michael Córdova Carrión	Ing. José Oswaldo Guamán Quinche

1. INTRODUCCIÓN

Este documento es una Especificación de Requisitos Software (ERS) para el Sistema de información sobre COVID-19 mediante llamadas telefónicas con la ayuda de un agente inteligente. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE 830.

1.1. Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales para el diseño del Sistema de información sobre COVID-19 mediante llamadas telefónicas con la ayuda de un agente inteligente”.

1.2. Alcance

La solución informática tiene como objetivo realizar el diseño de un sistema de información sobre COVID-19 mediante llamadas telefónicas, por tal motivo los requisitos están dirigida al usuario del sistema, para continuar con el desarrollo de la aplicación y profundizar en la automatización de ésta.

1.3. Personal involucrado

TABLA I

DESARROLLADOR DEL PROYECTO

Nombre	Sandro Michael Córdova Carrión
Rol	Analista, diseñador y programador
Categoría Profesional	Estudiante
Responsabilidad	Análisis de información, diseño y programación de la solución informática
Información de contacto	sandro.cordova@unl.edu.ec

TABLA II

DIRECTOR DEL TRABAJO DE TITULACIÓN

Nombre	José Oswaldo Guamán Quinche
Rol	Director del Trabajo de Titulación
Categoría Profesional	Ingeniero en Sistemas
Responsabilidad	Supervisar y asesorar en el desarrollo del Trabajo de Titulación
Información de contacto	Jose.o.guaman@unl.edu.ec

1.4. Definiciones, acrónimos y abreviaturas

TABLA III

DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

Nombre	Descripción
Usuario	Persona que usará el sistema
ERS	Especificación de Requisitos Software
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional

1.5. Referencias

TABLA IV

REFERENCIAS

Título del Documento	Referencia
Standard IEEE 830 - 1998	IEEE

1.6. Resumen

Este documento consta de tres secciones. En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda sección del documento se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en detalles excesivos.

Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requisitos que debe satisfacer el sistema.

2. DESCRIPCIÓN GENERAL

2.1. Perspectiva del producto

La solución informática será diseñada para trabajar en entornos web, lo que permitirá su utilización de forma rápida y eficaz.

2.2. Funcionalidad del producto

TABLA V

HISTORIA DE USUARIO 01

Historia de Usuario	
Número: 01	Usuario: Paciente
Nombre de la historia: Transferir llamada	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Programador responsable: Sandro Córdova	
Descripción: <ul style="list-style-type: none">Como paciente quiero que mi llamada sea transferida a uno de los recepcionistas telefónicos disponibles en el centro de salud.	
Validación: <ul style="list-style-type: none">Se asignará la llamada al recepcionista telefónico disponible.Fallo: Si no hay un recepcionista disponible, la llamada entra en espera.	

TABLA VI

HISTORIA DE USUARIO 02

Historia de Usuario	
Número: 02	Usuario: Sistema
Nombre de la historia: Generar cita médica	
Prioridad en negocio: media	Riesgo en desarrollo: Medio
Programador responsable: Sandro Córdova	
Descripción: <ul style="list-style-type: none">El sistema puede generar una cita médica para el paciente en caso de que los síntomas de este ameriten atención médica inmediata.	
Validación: <ul style="list-style-type: none">Información requerida: Número de cédula, nombres, apellidos.Cita médica generada: terminada la llamada.Cita no generada: se vuelve al menú principal.	

TABLA VII
HISTORIA DE USUARIO 03

Historia de Usuario	
Número: 03	Usuario: Sistema
Nombre de la historia: Clasificación del paciente.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Programador responsable: Sandro Córdova	
Descripción: <ul style="list-style-type: none"> El agente inteligente clasifica al paciente basado en los síntomas seleccionados. Clasificaciones posibles: Prioridad baja, prioridad media, prioridad alta. Prioridad baja: El agente inteligente brinda información general (HU4). Prioridad media: El agente inteligente brinda información general (HU04), es opcional la generación de una cita (HU2). Prioridad alta: El agente inteligente brinda información general (HU04), el agente inteligente genera una cita médica (HU2), el agente inteligente receta medicamentos (HU3). 	
Validación: <ul style="list-style-type: none"> Éxito: El agente inteligente realiza las acciones especificadas según la clasificación del paciente. Fallo: Se vuelve al menú principal. 	

TABLA VIII
HISTORIA DE USUARIO 04

Historia de Usuario	
Número: 04	Usuario: Sistema
Nombre de la historia: Medidas de bioseguridad	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Programador responsable: Sandro Córdova	
Descripción: <ul style="list-style-type: none"> El agente inteligente presentará al paciente audio guías con las medidas de bioseguridad que debe cumplir. 	

Validación:
<ul style="list-style-type: none"> • Información brindada: Se continúa con el proceso de información al usuario. • Fallo: Vuelve al menú principal.

TABLA IX
HISTORIA DE USUARIO 05

Historia de Usuario	
Número: 05	Usuario: Sistema
Nombre de la historia: Recetar en caso de emergencia	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Programador responsable: Sandro Córdova	
Descripción: <ul style="list-style-type: none"> • En caso de emergencia, el agente inteligente puede recetar al paciente medicamentos de emergencia como medida preventiva hasta que este pueda ser atendido en el centro de salud. 	
Validación: <ul style="list-style-type: none"> • Información brindada: Se continúa con el proceso de información al usuario. • Fallo: Vuelve al menú principal. 	

TABLA X
HISTORIA DE USUARIO 06

Historia de Usuario	
Número: 6	Usuario: Administrador
Nombre de la historia: Crear módulo	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Programador responsable: Sandro Córdova	
Descripción: <ul style="list-style-type: none"> • Como administrador quiero crear módulos para poder generar mis propios módulos. 	

Validación: <ul style="list-style-type: none"> Datos ingresados en el formulario correctamente validados: usuario y contraseña. Redirección a la interfaz de crear módulos. Mensaje “Modulo creado”. Fallo: mensaje de error.
Observaciones: <ul style="list-style-type: none"> El número de módulo debe ser único.

TABLA XI
HISTORIA DE USUARIO 07

Historia de Usuario	
Número: 7	Usuario: Administrador
Nombre de la historia: Visualizar módulos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Programador responsable: Sandro Córdova	
Descripción: <ul style="list-style-type: none"> Cómo administrador quiero visualizar los módulos creados para poder conocer el listado de los módulos. 	
Validación: <ul style="list-style-type: none"> Los módulos serán listados en el orden que se presentan al usuario. Se mostrará la información del módulo y opciones de editar, visualizar o eliminar. Fallo: mensaje de error en caso de no encontrar el módulo. 	

TABLA XII
HISTORIA DE USUARIO 08

Historia de Usuario	
Número: 8	Usuario: Administrador
Nombre de la historia: Actualizar módulos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio

Programador responsable: Sandro Córdova
Descripción: <ul style="list-style-type: none"> • Como administrador quiero editar los módulos creados para poder actualizar la información de éstas.
Validación: <ul style="list-style-type: none"> • Los datos en el formulario correctamente validados. • Mensaje afirmativo. • Fallo: mensaje de error si los campos del formulario están mal ingresados o si no se ha podido realizar la actualización.
Observaciones: <ul style="list-style-type: none"> • El módulo no se puede repetir con otros, este debe ser único.

TABLA XIII
HISTORIA DE USUARIO 09

Historia de Usuario	
Número: 9	Usuario: Administrador
Nombre de la historia: Borrar módulos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Programador responsable: Sandro Córdova	
Descripción: <ul style="list-style-type: none"> • Como administrador quiero eliminar los módulos creados para poder borrar definitivamente de la base de datos. 	
Validación: <ul style="list-style-type: none"> • Pedir confirmación de borrar datos mediante el mensaje “¿Desea eliminar el módulo?” • Mensaje de éxito “Módulo eliminado”. • Fallo: mensaje de error si el módulo está siendo utilizado por un bloqueo activo. 	
Observaciones: <ul style="list-style-type: none"> • Sólo se puede eliminar el módulo si no contiene lecciones en éste. 	

TABLA XIV
HISTORIA DE USUARIO 10

Historia de Usuario	
Número: 10	Usuario: Administrador
Nombre de la historia: Agregar cita	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Programador responsable: Sandro Córdova	
Descripción: <ul style="list-style-type: none"> Como administrador quiero agregar información de una cita al sistema para presentarlo a los usuarios. 	
Validación: <ul style="list-style-type: none"> Datos ingresados en el formulario correctamente validados: usuario y contraseña. Redirección a la interfaz de agregar información general Mensaje “Cita agregada”. Fallo: mensaje de error. 	
Observaciones: <ul style="list-style-type: none"> El número de módulo debe ser único. 	

TABLA XV
HISTORIA DE USUARIO 11

Historia de Usuario	
Número: 11	Usuario: Administrador
Nombre de la historia: Visualizar cita	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Programador responsable: Sandro Córdova	
Descripción: <ul style="list-style-type: none"> Como administrador quiero visualizar las citas presentadas por el sistema a los usuarios. 	
Validación: <ul style="list-style-type: none"> Las citas serán listadas en el orden que se presenta al usuario. Se mostrará la información general y opciones de editar o eliminar. 	

- Fallo: mensaje de error en caso de no encontrar el módulo.

TABLA XVI
HISTORIA DE USUARIO 12

Historia de Usuario	
Número: 12	Usuario: Administrador
Nombre de la historia: Actualizar cita	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Programador responsable: Sandro Córdova	
Descripción: <ul style="list-style-type: none"> • Cómo administrador quiero actualizar las citas presentadas por el sistema. 	
Validación: <ul style="list-style-type: none"> • Los datos en el formulario correctamente validados. • Mensaje afirmativo. • Fallo: mensaje de error si los campos del formulario están mal ingresados o si no se ha podido realizar la actualización. 	
Observaciones: <ul style="list-style-type: none"> • La información agregada no se puede repetir con otra, esta debe ser única. 	

TABLA XVII
HISTORIA DE USUARIO 13

Historia de Usuario	
Número: 13	Usuario: Administrador
Nombre de la historia: Borrar cita	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Programador responsable: Sandro Córdova	
Descripción: <ul style="list-style-type: none"> • Cómo administrador quiero eliminar las citas presentada por el sistema. 	

Validación:

- Solicitar confirmación para eliminar los datos mediante el mensaje “¿Desea eliminar la información?”
- Mensaje de éxito “Cita eliminada”.
- Fallo: mensaje de error.

2.3. Características de los usuarios

TABLA XVIII

CARACTERÍSTICAS DEL ADMINISTRADOR

Tipo de usuario	Administrador
Formación	Ninguna
Actividades	Control y manejo total del sistema en general.

TABLA XIX

CARACTERÍSTICAS DEL USUARIO

Tipo de usuario	Usuario
Formación	Ninguna
Actividades	<ul style="list-style-type: none">• Acceder a la información presentada por el sistema.• Acceder a las diferentes funcionalidades del sistema.• Solicitar atención personalizada.

2.4. Restricciones

- El sistema se desarrollará en base a la metodología XP por su adaptabilidad al Trabajo de Titulación.
- La aplicación será utilizada en un ordenador.
- El sistema deberá tener un diseño e implementación sencilla.
- Lenguajes y tecnologías en uso: Java, JADE.
- La aplicación web será desarrollada con un el framework Netbeans.

2.5. Suposiciones y dependencias

- Se asume que los requisitos descritos en el presente documento son estables.
- Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados para garantizar una ejecución correcta de la misma.

3. REQUISITOS ESPECÍFICOS

3.1. Requisitos comunes de las interfaces

3.1.1. Interfaces de usuario

La interfaz con el usuario consistirá en una serie de opciones planteadas por el sistema en donde, el usuario podrá seleccionar cualquiera de estas opciones mediante su voz. Durante la interacción sistema-usuario la comunicación se da únicamente mediante llamada telefónica desde cualquier dispositivo móvil o fijo.

3.1.2. Interfaces de hardware

Será necesario disponer de lo siguiente:

- Computador.
- Teléfono fijo.
- Teléfono móvil.

3.1.3. Interfaces de software

- Sistema Operativo: Windows XP o superior.

3.1.4. Interfaces de comunicación

El computador se comunicará con el teléfono fijo encargado de recepcionar las llamadas haciendo uso del puerto SERIAL.

3.2. Requerimientos funcionales

En el presente apartado se detalla los requisitos funcionales que serán utilizados en el sistema.

TABLA XX
REQUERIMIENTO FUNCIONAL 01

Identificación del requisito	RF01
Nombre del requisito	Transferir llamada
Descripción del requisito	El usuario puede solicitar al sistema comunicación directa con un departamento de atención al cliente dentro del centro de salud.
Prioridad del requisito	Media

TABLA XXI
REQUERIMIENTO FUNCIONAL 02

Identificación del requisito	RF02
Nombre del requisito	Listar síntomas
Descripción del requisito	El sistema permitirá al usuario listar los síntomas que este posee para poder determinar el estado del paciente.
Prioridad del requisito	Alta

TABLA XXII
REQUERIMIENTO FUNCIONAL 03

Identificación del requisito	RF03
Nombre del requisito	Clasificación del estado de los pacientes
Descripción del requisito	El sistema compara y clasifica los síntomas dados por el paciente y toma decisiones dependiendo de la clasificación.
Prioridad del requisito	Alta

TABLA XXIII
REQUERIMIENTO FUNCIONAL 04

Identificación del requisito	RF04
Nombre del requisito	Informar sobre medidas de bioseguridad
Descripción del requisito	El sistema debe presentar al usuario las medidas de bioseguridad necesarias para poder acercarse al centro de salud.
Prioridad del requisito	Media

TABLA XXIV
REQUERIMIENTO FUNCIONAL 05

Identificación del requisito	RF05
-------------------------------------	------

Nombre del requisito	Agendar citas
Descripción del requisito	El usuario podrá agendar una cita médica a través del sistema. Los datos para agendar una cita son: Especialidad médica, doctor, número de cédula, nombres, apellidos, fecha y hora.
Prioridad del requisito	baja

TABLA XXV

REQUERIMIENTO FUNCIONAL 06

Identificación del requisito	RF06
Nombre del requisito	Recetar en caso de emergencia
Descripción del requisito	En caso de emergencia, el sistema debe recomendar al usuario el consumo exclusivo de Paracetamol, hasta que este pueda ser atendido en el centro de salud.
Prioridad del requisito	Alta

TABLA XXVI

REQUERIMIENTO FUNCIONAL 07

Identificación del requisito	RF07
Nombre del requisito	Tiempo de interacción 6min
Descripción del requisito	El tiempo máximo de interacción entre el sistema y el usuario debe ser no mayor a seis minutos.
Prioridad del requisito	Media

Tabla XXVII

REQUERIMIENTO FUNCIONAL 08

Identificación del requisito	RF08
Nombre del requisito	Voz aguda y agradable

Descripción del requisito	La voz que debe tener el sistema al comunicarse con el usuario debe ser una voz aguda y agradable al oído.
Prioridad del requisito	Baja

TABLA XXVIII
REQUERIMIENTO FUNCIONAL 09

Identificación del requisito	RF09
Nombre del requisito	Gestión de módulos
Descripción del requisito	El administrador podrá crear, visualizar, actualizar y borrar módulos. Para poder gestionar módulos el administrador debe autenticarse mediante usuario y contraseña.
Prioridad del requisito	Alta

TABLA XXIX
REQUERIMIENTO FUNCIONAL 10

Identificación del requisito	RF10
Nombre del requisito	Gestión de información general
Descripción del requisito	El administrador podrá crear, visualizar, actualizar y borrar información general del centro de salud. Para poder gestionar la información general el administrador debe autenticarse mediante usuario y contraseña.
Prioridad del requisito	Alta

TABLA XXX
REQUERIMIENTO FUNCIONAL 11

Identificación del requisito	RF11
Nombre del requisito	Gestión de citas

Descripción del requisito	El administrador podrá crear, visualizar, actualizar y borrar la información de las citas médicas del centro de salud. Para poder gestionar las citas médicas el administrador debe autenticarse mediante usuario y contraseña.
Prioridad del requisito	Alta

TABLA XXXI
REQUERIMIENTO FUNCIONAL 12

Identificación del requisito	RF12
Nombre del requisito	Generar reportes
Descripción del requisito	El administrador podrá visualizar la información referente a las llamadas que se han realizado al centro de salud, pacientes que han recibido medidas de seguridad y citas médicas generadas.
Prioridad del requisito	media

3.3. Requerimientos no funcionales

Tabla XXXII
REQUISITOS NO FUNCIONALES

Identificación	Atributo	Descripción
RNF01	Rendimiento	El sistema debe garantizar que las consultas u otro proceso no afecte el desempeño de la base de datos, ni considerablemente el tráfico de la red.
RNF02	Seguridad	<p>El sistema contará con autenticación de administrador mediante el usuario y contraseña asignados.</p> <p>El sistema garantizará la seguridad con respecto a la información y datos personales que el usuario proporcione al momento de generar una cita médica.</p>

RNF03	Fiabilidad	El sistema deberá contar con una interfaz de uso intuitiva y sencilla.
RNF04	Disponibilidad	La disponibilidad del sistema deberá ser continua con un nivel de servicio para los usuarios de 24 horas del día durante los 365 días del año, con un estimado de inactividad para tareas de mantenimiento y fallos no planificados de 1 días.
RNF05	Mantenibilidad	El sistema debe disponer de una documentación fácilmente actualizable que permita realizar operaciones de mantenimiento.
RNF06	Portabilidad	Es sistema deberá ser compatible con la mayoría de los sistemas operativos en los que se ejecute.
RNF07	Usabilidad	El sistema debe presentar una secuencia ordenada de las opciones que el usuario pueda seleccionar durante su interacción con el sistema.

C. Anexo III

Elicitación de requerimientos

En este proceso se presenta el proceso de elicitación de requerimientos para la captura de los requisitos de sistema de software. A continuación, se describe la adquisición de requisitos por medio de la técnica de la entrevista, el análisis y validación de los requerimientos finales.

1. Entrevista

La entrevista se realizó a la/las personas a cargo del departamento de coordinación, que son quienes se encargan de la atención personal y telefónica de los pacientes de los diferentes establecimientos de asistencia y tratamiento médico en la ciudad de Loja. A continuación, se listan los nombres de los establecimientos médicos:

- Sueanny Vélez, "Consorcio médico" Atención al cliente (PE01)
- Ruth Ojeda, "Clínica hospital municipal 'JULIA ESTHER GONZÁLEZ DELGADO'" Departamento de Coordinación (PE02)
- Mary Orellana, "Clínica hospital municipal 'JULIA ESTHER GONZÁLEZ DELGADO'" Departamento de farmacia (PE03)
- Alexandra Quizhpe, "Clínica Abendaño" Departamento de coordinación (PE04)
- Carolina Guerrero, "Clínica Abendaño" Departamento de pediatría (PE05)
- María Fernanda Celi, "Clínica Mogrovejo" Departamento de coordinación (PE06)
- Jessica Ruiz, "Clínica San José" Departamento de coordinación (PE07)

El objetivo de la entrevista es obtener información sobre el proceso seguido cuando ingresa una llamada al centro de salud, además de información básica que le interesa que se brinde al usuario, recomendaciones generales y otros problemas que le gustaría que esta solucione.

PE: Persona Entrevistada

RP: Respuesta Pregunta

1.1 (RP01) ¿Cuál es el proceso que se sigue cuando una persona llama a la institución en busca de información o soporte médico?

PE01:

No hay un protocolo definido, los pacientes preguntan directamente por un médico.

PE02:

Se recibe directamente la llamada, y luego se la asigna a un departamento.

PE03:

No se recibe directamente la llamada, él que tiene el protocolo es en coordinación que es donde receptan la llamada y luego se la asigna a este departamento.

PE04:

Asignación a la secretaria de cada médico mediante call center para que ella le ayude al paciente.

PE05:

- Costo
- Agendar
- Re agenda la cita

PE06:

- Solicitar cita médica.
- Pedir el número de teléfono de algún doctor si es una emergencia.
- Costos.

EP07:

1.2 (RP02) ¿El Proceso seguido es el definido por la institución o se sigue un estándar general para atención médica?

PE01:

El establecido por la clínica.

PE02:

Es por la institución y se cumplen con las normas del ministerio de salud pública.

PE03:

Es por la institución y se cumplen con las normas del ministerio de salud pública.

PE04:

El establecido por la clínica.

PE05:

El establecido por la clínica.

PE06:

El establecido por la clínica.

PE07:

El establecido por la clínica.

1.3 (RP03) ¿Qué opciones tiene la persona una vez que se comunica mediante llamada telefónica con la institución?

PE01:

- Reservas de citas.
- Precio de la consulta de los doctores dependiendo la especialidad.
- Agenda citas para consultas médicas.

PE02:

- Consultas de precios.
- Horarios de atención.
- Asignación o reserva de turnos.
- Exámenes de laboratorio.
- Se maneja por diferentes departamentos.

PE03:

Como es un área farmacología, directamente con nosotros no se comunican para ese tipo de información, pero si es que se diera el caso se da la información de horarios, atención médica y por supuesto cuando se trata de algún medicamento en general

PE04:

Asignación de la llamada a cada uno de los asistentes.

PE05:

- Costo.
- Agendar.
- Re agenda la cita.

PE06:

- Solicitar cita médica.
- Pedir el número de teléfono de algún doctor si es una emergencia.
- Costos.

PE07:

- Con que doctor desea atenderse
- El precio del doctor
- Horarios de doctores
- Asignación de turnos

1.4 (RP04) Si la sintomatología del paciente o el motivo de la llamada es por contagio de COVID-19. ¿Se acepta en la institución este tipo de pacientes?

PE01:

Para ingresos en la clínica se maneja, en el consorcio solo los pacientes que posiblemente presentan COVID-19 con el oftalmólogo, el doctor hace las pruebas para confirmar un diagnóstico y se los acepta siguiendo el debido protocolo de bioseguridad.

PE02:

No se aceptan pacientes con COVID-19.

PE03:

Realmente nosotros no tenemos ningún caso COVID-19, de ser el caso tendrían que comunicarse directamente con el médico, no se recomienda ningún medicamento.

PE04:

Claro, no se puede negar la atención a ningún paciente.

PE05:

Si.

PE06:

No, solo es estudio de tomografía, pero no para hospitalización.

PE07:

Si.

1.1 (RP05) ¿Cuáles son las recomendaciones que se brinda a las personas que se comunican con síntomas de COVID-19?

PE01:

- Que vengán cumpliendo las medidas de bioseguridad, si es posible con doble mascarilla, que mantengan el distanciamiento y esperen a su doctor.
- Se siguen todas las medidas de bioseguridad establecidas.
- Solo con un acompañante.

PE02:

Solo en el caso de que tengan alguna duda de algún medicamento que les hayan recetado, se le comunica con el médico que le dio la receta, pero no se atiende a personas con COVID-19, se les recomienda que vayan a una casa de salud donde asistan pacientes con COVID-19.

PE03:

Nosotros como farmacia interna de la clínica receptamos recetas médicas, si es que el caso de un paciente que viniera, lo que nosotros podemos ayudar es reservar un turno con el médico y es él facultativo quien determina que pasos puede seguir.

PE04:

El médico que atiende el COVID-19 tiene su secretaria personal, pero se solicita traje de bioseguridad, alcohol, doble mascarilla.

PE05:

Las medidas de bioseguridad que están establecidas, visor, distanciamiento, mascarilla y lavado de manos uso de alcohol, se le restringe a un acompañante, se receta solo paracetamol.

PE06:

Venir de preferencia solo o acompañado de un solo familiar, traer mascarilla NK95 y un protector facial, venir en horas donde no haya mucha afluencia de gente, en horarios de la tarde.

PE07:

Que vengan protegidos, con mascarilla, si han estado en contacto con otros familiares para que también se realicen la prueba y aislamiento.

1.2 (RP06) ¿Cuáles son las dudas por las que generalmente se comunican las personas con el centro de salud?

PE01:

Para agendar citas con los diferentes especialistas del consorcio médico.

PE02:

- Horarios de atención de los médicos.
- Preguntando los servicios que ofrece la casa de salud.

PE03:

- Dudas con respecto a farmacia.
- Medicamento que se puede administrar con otro nombre comercial o genérico.
- Otros medicamentos que tengan la misma composición sirva para el mismo síntoma o enfermedad.

PE04:

- Para saber de determinado médico y que especialidad tiene.
- El horario.
- Costo de consulta.
- Información general.

PE05:

- Si sus niños están enfermos tienen alguna patología.
- Si necesitan de algún tratamiento.
- Si ya finalizan el tratamiento.
- Preguntando costos de la consulta.

PE06:

Agendar turnos.

PE07:

El valor que cobran.

1.3 (RP07) Piensa usted que sería de gran ayuda un sistema de agentes inteligentes que brinde al paciente este tipo de recomendaciones y solucione dudas comunes, dejando así solo aquellas llamadas de personas que realmente necesitan asistencia por parte del personal del centro de salud.

PE01:

PE02:

Si, nos ahorraría tiempo.

PE03:

Por el momento no le vemos necesario ya que solo se maneja directamente farmacia.

PE04:

Es complicado ya que cada médico tiene horarios diferentes.

PE05:

Sí, sería importante para optimizar tiempo de pacientes.

PE06:

Sí, sería necesario, útil y así poder brindar un trato o asistencia mejor a los pacientes que vienen.

PE07:

Ayudaría informando horarios y agendando turnos.

1.4 (RP08) Si este tipo de tecnología se aplica al centro de salud:

PE01:

- a. **¿Qué otra funcionalidad se le podría añadir?**
- b. **¿Cuál sería el tiempo estimado para la interacción entre el sistema que gestione la llamada y el paciente?**
Generalmente un minuto máximo tres
- c. **¿La voz que escucharía el paciente al momento de comunicarse debe ser masculina o femenina? ¿Grave o aguda?**
Creo que femenina, suave

PE02:

- a. **¿Qué otra funcionalidad se le podría añadir?**
- b. **¿Cuál sería el tiempo estimado para la interacción entre el sistema que gestione la llamada y el paciente?**
Unos dos minutos máximo 4 minutos
- c. **¿La voz que escucharía el paciente al momento de comunicarse debe ser masculina o femenina? ¿Grave o aguda?**
Femenina, grave

PE03:

- a. **¿Qué otra funcionalidad se le podría añadir?**
Sería más un sistema interno.
- b. **¿Cuál sería el tiempo estimado para la interacción entre el sistema que gestione la llamada y el paciente?**
Generalmente de dos minutos y máximo hasta que el paciente pueda despejar su duda.
- c. **¿La voz que escucharía el paciente al momento de comunicarse debe ser masculina o femenina? ¿Grave o aguda?**
Debe ser una voz tranquila, serena sin importar si es masculina o femenina.

PE04:

- a. **¿Qué otra funcionalidad se le podría añadir?**
No, se me ocurre nada.
- b. **¿Cuál sería el tiempo estimado para la interacción entre el sistema que gestione la llamada y el paciente?**
Depende puede ser de un minuto, hasta diez.
- c. **¿La voz que escucharía el paciente al momento de comunicarse debe ser masculina o femenina? ¿Grave o aguda?**
Voz grave.

PE05:

- a. **¿Qué otra funcionalidad se le podría añadir?**
Clasificar a los pacientes, y a que consulta va a ir.
- b. **¿Cuál sería el tiempo estimado para la interacción entre el sistema que gestione la llamada y el paciente?**
No más de treinta segundos.
- c. **¿La voz que escucharía el paciente al momento de comunicarse debe ser masculina o femenina? ¿Grave o aguda?**
Masculina, grave.

PE06:

- a. **¿Qué otra funcionalidad se le podría añadir?**
Cirugías para que los doctores puedan agendarlas.
- b. **¿Cuál sería el tiempo estimado para la interacción entre el sistema que gestione la llamada y el paciente?**
Tres minutos.
- c. **¿La voz que escucharía el paciente al momento de comunicarse debe ser masculina o femenina? ¿Grave o aguda?**
Tono de voz aguda.

PE07:

- a. **¿Qué otra funcionalidad se le podría añadir?**
Enviándoles un mensaje cuando les toque la cita para confirmación.
- b. **¿Cuál sería el tiempo estimado para la interacción entre el sistema que gestione la llamada y el paciente?**
5 min máximo 20 min.
- c. **¿La voz que escucharía el paciente al momento de comunicarse debe ser masculina o femenina? ¿Grave o aguda?**
No afectaría solo sería un buen trato una voz normal.

2. Lista Preliminar de requisitos

N°	Requisito	Solicitado/ Inferido	Referencia
01	El sistema debe permitir comunicarse con la secretaría de un médico en específico	Inferido	RP01: PE04
02	El sistema debe permitir comunicarse con un departamento específico	Inferido	RP01: PE03, PE02
03	El sistema debe permitir agendar citas	Inferido	RP01: PE05, PE06
04	El sistema debe permitir consultar el precio de la consulta médica	Inferido	RP3: PE01
05	El sistema debe permitir acceder al número de teléfono de los médicos	Inferido	RP3: PE06
06	El sistema debe tener un apartado para pacientes con COVID-19	Inferido	RP04: PE01, PE03, PE04, PE05, PE07
07	El sistema debe informar si el centro de salud admite pacientes con COVID-19	Inferido	RP04: PE01, PE03, PE04, PE05, PE07
08	El sistema debe informar sobre las medidas de bioseguridad a cumplir el paciente para poder asistir al centro de salud.	Inferido	RP05: PE01, PE02, PE03, PE04, PE05, PE06, PE07
09	El sistema solo puede recetar paracetamol en caso de requerir algún medicamento	Inferido	RP05: PE05
10	El tiempo máximo de iteración entre el paciente y el sistema debe ser de 6min	Inferido	RP08: PE01 - PE07(b)
11	El sistema debe informar con un tono de voz agudo y agradable	Solicitado	RP08: PE01(c), PE03(c), PE06(c), PE07(c)
12	El administrador podrá gestionar la información general del centro de salud. (crear, consultar, modificar, buscar y eliminar)	Inferido	RP01: PE01
13	El administrador podrá gestionar la información de las citas médicas, como precios, días, horarios (crear, consultar, modificar, buscar y eliminar).	Inferido	RP03: PE01
14	El administrador podrá gestionar la información cada uno de los médicos que trabajen en el centro de salud como	Inferido	RP01: PE04 RP03: PE06

	disponibilidad y números de teléfono (crear, consultar, modificar, buscar y eliminar).		
15	El administrador podrá gestionar módulos del sistema (crear, consultar, modificar, buscar y eliminar)	Inferido	

Tabla 1 Lista preliminar de requisitos

3. Matriz de iteración

Por medio de la matriz de iteración se evalúa los requisitos para identificar conflictos o solapamientos. La matriz presenta dos entradas, donde cada entrada contiene todos los requisitos, obteniendo que estos relacionen entre sí.

- Solapamiento: Si entre r_1 y r_2 tratan los mismos aspectos del sistema, dando redundancia. Se lo representa con la letra S.
- Conflicto: Si entre r_1 y r_2 son contradictorios, dando problemas de consistencia interna. Se lo representa con la letra C.

En la tabla 2 se presenta el solapamiento o conflicto que puede existir entre los requisitos anteriormente obtenidos.

R	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
01					C										
02															
03															
04															
05	C														
06							C								
07						C									
08															
09															
10															
11															
12															
13															
14															
15															

Tabla 2 Matriz de iteración

La descripción de los conflictos encontrados en la matriz realizada se presenta en la Tabla 3. **Error! No se encuentra el origen de la referencia.**, con ello se puede hacer un análisis y brindar una solución al requisito en conflicto por medio de la creación de un nuevo requisito.

Solapamiento y conflictos

N°	Requisito	Error	Descripción	Requisito final
1	01, 05	C	El requisito 01 tiene el mismo propósito del requisito 05.	El sistema debe permitir comunicarse con un médico en específico (número personal o la extensión de su secretaria).
2	06, 07	C	La negación del requisito 07 contradice al requisito 06	El sistema debe tener un apartado para pacientes con COVID-19

Tabla 3 Solapamiento y conflictos

4. Lista final de requisitos

Una vez solucionado los conflictos encontrados, en la Tabla 4 se presenta los requisitos finales que serán aplicados en el desarrollo de la aplicación web.

Código	Requisitos
01	Comunicación con los médicos
02	Comunicación los diferentes departamentos
03	Agendar citas
04	Informar costos
05	Apartado para pacientes con COVID-19
06	Informar sobre medidas de bioseguridad
07	Recetar en caso de emergencia
08	Tiempo de iteración de 6min
09	Voz aguda y agradable
10	Gestión de módulos
11	Gestión de información general
12	Gestión de citas
13	Gestión de médicos

Tabla 4 Lista final de requisitos

5. Entrevistados



Carrera de Ingeniería en
Sistemas / Computación



Universidad
Nacional
de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

Nombre	Teléfono	Cédula	Firma
Henio Fernando Kella	0967673345	1104642531	
Carolina Gueña	0987209691	1150131785	
Alexandra Ruiz	1103856199 0970187416	1103856199	
Jessica Ruiz	0982462722	1104545429	
Ruth Ojeda	0983028641	1102583392	
Mary Orellana	0983395913	1103826189	
Seanny Velaz	0995774420	1104216765	











D. Anexo IV

A continuación se muestra el documento en donde se detalla la arquitectura del software planteado.

Documento de Arquitectura de Software

Diseño de un sistema de información sobre COVID-19 mediante
llamadas telefónicas con la ayuda de un agente inteligente

1. Introducción

Este documento es un resumen general sobre la arquitectura del software para el Diseño de un sistema de información sobre COVID-19 mediante llamadas telefónicas con la ayuda de un agente inteligente. Este documento se ha estructurado basándose en el modelo 4+1, en donde se utiliza las vistas necesarias de arquitectura para describir los aspectos del sistema.

1.1. Propósito

El presente documento tiene como propósito describir el diseño de la arquitectura de software para el Diseño de un sistema de información sobre COVID-19 mediante llamadas telefónicas con la ayuda de un agente inteligente a través del modelo 4+1.

1.2. Alcance

El documento de Arquitectura abarca la definición de la arquitectura de la aplicación web a través de las vistas de lógica, de despliegue, de procesos, física y de escenarios.

1.3. Referencias

TABLA A5 I.
REFERENCIAS

Título del Documento	Referencia
Arquitectura de Software 4+1	Modelo
Especificación de Requerimientos de Software IEEE 830	Anexo 2

1.4. Resumen

Este documento consta de cuatro secciones. En la primera sección se realiza una introducción al mismo y se proporciona una visión general del documento de Arquitectura de Software.

En la segunda sección del documento se realiza una descripción general de la Arquitectura, con el fin de conocer el modelo 4+1 y los objetivos.

En la tercera sección del documento se definen las vistas y entregables de acuerdo al modelo 4+1, que son: vista de escenarios (diagrama de Casos de Uso), vista lógica (Diagrama de Clases), vista de despliegue (Diagrama de Componentes), vista de procesos (Diagrama de Actividad) y vista física (Diagrama de Despliegue).

Por último, la cuarta sección del documento es aquella en la que se define la Arquitectura de la aplicación.

2. Descripción General

2.1. Representación de la Arquitectura

La solución informática utiliza una arquitectura cliente servidor, a través del modelo 4+1, misma que propone cuatro vistas bien diferenciadas y que se relacionan entre sí con una vista más, que se denomina vista “+1” o vista de escenarios. En la TABLA A5 II se describe cada una de las vistas.

TABLA A5 II.
ARQUITECTURA 4+1.

Vista	Perspectiva	Elemento	Descripción
Vista Lógica	<ul style="list-style-type: none">• Usuario Final• Programador	<ul style="list-style-type: none">• Diagrama de Clases• Diagrama de base de datos	Representa la funcionalidad que el sistema proporcionara a los usuarios finales.
Vista de Despliegue	<ul style="list-style-type: none">• Administrador de Software• Programador	<ul style="list-style-type: none">• Diagrama de Paquetes	Se muestra cómo está dividido el sistema software en componentes y las dependencias que hay entre esos componentes.
Vista Física	<ul style="list-style-type: none">• Ingeniero de Sistemas	<ul style="list-style-type: none">• Diagrama de Despliegue	Se muestra todos los componentes físicos del sistema, así como todas las conexiones físicas entre esos componentes que conforman la solución (incluyendo los servicios).
Vista de Procesos	<ul style="list-style-type: none">• Integrador de Sistemas	<ul style="list-style-type: none">• Diagrama de Actividad	Se muestra los procesos que hay en el sistema y la forma en la que se comunican estos procesos.
Vista de Escenarios (+1)		<ul style="list-style-type: none">• Diagrama de Casos de Uso	Esta vista va ser representada por los casos de uso, teniendo

			como objetivo unir y relacionar las otras 4 vistas.
--	--	--	---

2.2. Objetivos de la arquitectura

El desarrollo del sistema cumplirá con las siguientes características:

- **Fiabilidad:** Esta vista va ser representada por los casos de uso, teniendo como objetivo unir y relacionar las otras 4 vistas.
- **Disponibilidad:** La disponibilidad del sistema deberá ser continua con un nivel de servicio para los usuarios de 24 horas del día durante los 365 días del año
- **Funcionabilidad:** El sistema debe responder al usuario en menos de 3 a 4 segundos.
- **Rendimiento:** La aplicación web proporcionara de forma rápida y precisa la información solicitada por el usuario.
- **Seguridad:** El sistema contará con autenticación de usuario mediante el correo electrónico y contraseña; además garantizará la seguridad con respecto a la información y datos que se mantengan (correo electrónico, contraseña, nombre).
- **Usabilidad:** El sistema debe presentar una interfaz de ayuda para que los usuarios se les facilite el uso del sistema.

3. Modelo 4+1

3.1. Vista de Escenarios

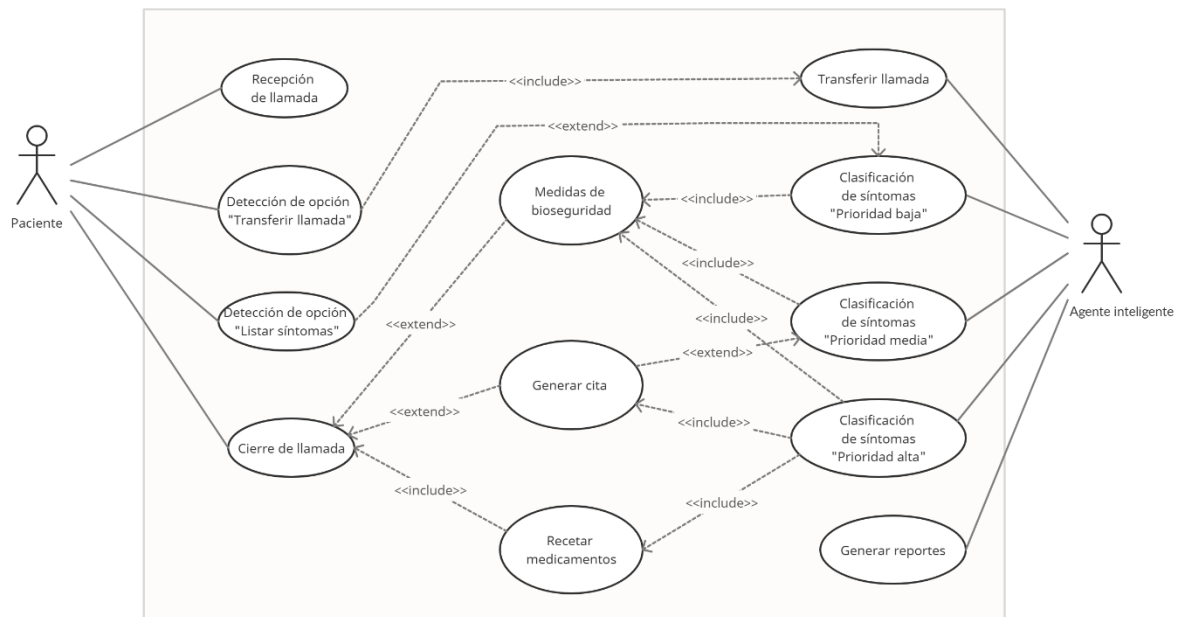
En esta vista se representada los casos de uso, donde se muestra la interacción de los diferentes actores con el sistema, teniendo como objetivo unir y relacionar las otras 4 vistas.

3.1.1. Actores del sistema

Identificador	AS-01
Nombre	Paciente
Descripción	Representa al usuario que realiza la llamada y accede al centro de llamadas inteligente. Se asume que la persona que se contacta con el centro de llamadas es el paciente de forma directa o indirecta. Generalmente este usuario puede hacer que su llamada se transfiera al recepcionista telefónico o ser atendido por el agente inteligente.

Identificador	AS-02
Nombre	Agente inteligente
Descripción	Representa al sistema. Se encarga de transferir la llamada del paciente (AS-01), informar y brindar soluciones efectivas al paciente acorde a la información recibida.

3.1.2. Diagrama de casos de uso



3.1.3. Especificación de Casos de Uso

Identificador	UC-01
Nombre	Recepción de llamada
Descripción	El paciente se comunica con centro de llamadas inteligente
Actores	Paciente, sistema
Precondiciones	El paciente debe realizar la llamada al número de la línea designado al centro de llamadas inteligente.
Postcondiciones	La llamada es asistida por el agente inteligente designado.
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El paciente se comunica con el centro de salud mediante llamada telefónica. 2. El sistema asigna un agente inteligente 3. El agente inteligente presenta al usuario la opción de “Transferir llamada” (CU-02). 4. El agente inteligente presenta al usuario la opción de “Listar síntomas” (CU-04). 	

Flujos alternos y excepciones	
1A. El paciente cierra la llamada	
2A. El sistema deshabilita el agente inteligente designado a la llamada.	

Identificador	UC-02
Nombre	Detección de opción “Transferir llamada”
Descripción	Si el paciente lo solicita, el agente inteligente transfiere la llamada a la línea telefónica designada al recepcionista telefónico del centro de salud.
Actores	Paciente, agente inteligente
Precondiciones	El sistema asignó un agente inteligente para asistir la llamada.
Postcondiciones	Se da inicio al caso de uso “Transferir llamada” (CU-03).
Flujo normal de eventos	
1A. El paciente solicita mediante voz que su llamada sea transferida.	
2A. El agente inteligente inicia el caso de uso “Transferir llamada” (CU-03).	
Flujos alternos y excepciones	
1A. El paciente se mantiene en la llamada a espera de la opción “listar síntomas” (CU-04)	

Identificador	UC-03
Nombre	Transferir llamada
Descripción	El agente inteligente transfiere la llamada a la línea designada al recepcionista telefónico del centro de salud.
Actores	Agente inteligente
Precondiciones	El paciente solicitó que su llamada sea transferida.
Postcondiciones	La llamada del paciente paciente es transferida a la línea designada al recepcionista telefónico del centro de salud.
Flujo normal de eventos	
1.1. El agente inteligente transfiere la llamada a la línea telefónica preestablecida.	
2.1. El sistema deshabilita el agente inteligente designado a la llamada.	
Flujos alternos y excepciones	
1A. (Si la línea telefónica preestablecida está ocupada o deshabilitada) El agente inteligente presenta al paciente las opciones iniciales (CU-02).	

Identificador	UC-04
Nombre	Detección de opción “Listar síntomas”
Descripción	El paciente tiene la opción de brinda una lista de forma verbal con los síntomas que posee.
Actores	Paciente
Precondiciones	El paciente no ha seleccionado la opción de “transferir llamada (CU-02)
Postcondiciones	El agente realiza una clasificación del estado del paciente basado en sus síntomas (CU-05, CU-06, CU-07).
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El paciente se mantiene en la línea. 2. El agente inteligente brinda las indicaciones para listar los síntomas que posee el paciente. 3. El paciente lista los síntomas de forma verbal. 4. El paciente mantiene silencio por 10s. 5. Se da inicio al caso de uso “prioridad baja” 	
Flujos alternos y excepciones	
<ol style="list-style-type: none"> 5.1 Se da inicio al caso de uso “prioridad media”. 5.2 Se da inicio al caso de uso “prioridad alta”. 	

Identificador	UC-05
Nombre	Clasificación de síntomas “prioridad baja”
Descripción	El agente inteligente compara y clasifica los síntomas dados por el paciente como un caso de prioridad baja.
Actores	Agente inteligente
Precondiciones	El paciente brindó una lista de síntomas.
Postcondiciones	Se brinda al usuario las medidas de bioseguridad que debe acatar (CU-08).
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El agente inteligente identifica los síntomas listados por el paciente. 2. El agente inteligente compara los síntomas listados con los almacenados en la base de datos. 3. El agente inteligente determina si los síntomas listados pertenecen a la clasificación “prioridad baja”. 4. Inicia el caso de uso “medidas de bioseguridad” (CU-08). 	

Flujos alternos y excepciones	
3A.	El agente inteligente determina si los síntomas listados pertenecen a la clasificación “prioridad media”.
4A.	Inicia el caso de uso “Clasificación de síntomas ‘prioridad media” (CU-06).

Identificador	UC-06
Nombre	Clasificación de síntomas “prioridad media”
Descripción	El agente inteligente compara y clasifica los síntomas del paciente como un caso de prioridad media.
Actores	Agente inteligente
Precondiciones	El paciente brindó una lista de síntomas.
Postcondiciones	Se brinda al usuario las medidas de bioseguridad que debe acatar (CU-08) y si este lo desea puede generar una cita (CU-09).

Flujo normal de eventos	
1.	El agente inteligente identifica los síntomas listados por el paciente.
2.	El agente inteligente compara los síntomas listados con los almacenados en la base de datos.
3.	El agente inteligente determina si los síntomas listados pertenecen a la clasificación “prioridad media”.
4.	Inicia el caso de uso “medidas de bioseguridad” (CU-08).
Flujos alternos y excepciones	
5A.	Inicia el caso de uso “Generar cita” (CU-09).
3B.	El agente inteligente determina si los síntomas listados pertenecen a la clasificación “prioridad alta”.
4B.	Inicia el caso de uso “Clasificación de síntomas ‘prioridad alta” (CU-07).

Identificador	UC-07
Nombre	Clasificación de síntomas “prioridad alta”
Descripción	El agente inteligente compara y clasifica los síntomas del paciente como un caso de prioridad alta.
Actores	Agente inteligente
Precondiciones	El paciente brindó una lista de síntomas.

Postcondiciones	Se brinda al usuario las medidas de bioseguridad que debe acatar (CU-08), el agente inteligente genera una cita (CU-09) y receta medicamentos preventivos (CU-10).
Flujo normal de eventos	
1. El agente inteligente identifica los síntomas listados por el paciente. 2. El agente inteligente compara los síntomas listados con los almacenados en la base de datos. 3. El agente inteligente determina si los síntomas listados pertenecen a la clasificación “prioridad media”. 4. Inicia los casos de uso, “generar cita” (CU-09), “medidas de bioseguridad” (CU-08) y “recetar medicamentos” (CU-10).	
Flujos alternos y excepciones	
3A. El agente inteligente determina que los datos brindados no pertenecen a ninguna de las clasificaciones. 4A. El agente inteligente brinda al paciente las opciones iniciales (CU-02).	

Identificador	UC-08
Nombre	Medidas de bioseguridad
Descripción	En caso de que los síntomas del paciente lo ameriten, el agente inteligente brinda al paciente audio guías con las medidas de bioseguridad que debe cumplir.
Actores	Agente inteligente
Precondiciones	El agente inteligente clasificó los síntomas del paciente en uno de los tres niveles de prioridad.
Postcondiciones	El paciente recibirá audio guías con las medidas de bioseguridad.
Flujo normal de eventos	
1B. El agente inteligente obtiene las audio guías que serán presentadas al paciente. 2B. El agente inteligente presenta las audio guías al paciente. 3B. El agente inteligente termina la llamada (CU-11).	
Flujos alternos y excepciones	
3A. El agente inteligente genera una cita (CU-09). 1A. El agente inteligente receta medicamentos (CU-10).	

Identificador	UC-09
Nombre	Generar cita
Descripción	En caso de que los síntomas del paciente lo ameriten y si el paciente desea, el agente inteligente genera una cita médica con los datos personales del paciente.
Actores	Agente inteligente
Precondiciones	El agente inteligente clasificó los síntomas del paciente en uno de los tres niveles de prioridad.
Postcondiciones	El paciente recibirá fecha de la cita para poder atenderse en el centro de salud.
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El agente inteligente solicita al paciente que se ingresen sus datos personales (Nombres y apellidos, edad, número de cédula y correo). 2. El paciente brinda al agente inteligente mediante voz sus datos personales. 3. El agente inteligente solicita confirmación de los datos personales brindados por el paciente. 4. El paciente confirma la información brindada. 5. El agente inteligente genera una cita para el paciente. 6. El agente inteligente informa al paciente la fecha de la cita. 7. El agente inteligente termina la llamada (CU-11). 	
Flujos alternos y excepciones	
<p>4A. El paciente rechaza la información brindada.</p> <p>5A. El flujo de eventos se reinicia.</p> <p>4B. El agente inteligente presenta al paciente las medidas de bioseguridad que debe cumplir (CU-08).</p>	

Identificador	UC-10
Nombre	Recetar medicamentos
Descripción	En el caso de que los síntomas del paciente lo ameriten y si el estado físico o médico del paciente lo permite, el agente inteligente indica al paciente los medicamentos que puede tomar de manera preventiva.
Actores	Agente inteligente
Precondiciones	El agente inteligente clasificó los síntomas del paciente como "prioridad alta".

Postcondiciones	El paciente recibirá la lista de medicamentos preventivos.
Flujo normal de eventos	
1. El agente inteligente obtiene la información que será presentada al paciente. 2. El agente inteligente presenta al paciente mediante audio la lista de medicamentos de los que puede hacer uso. 3. El agente inteligente cierra la llamada (CU-11).	
Flujos alternos y excepciones	
3A. El paciente cierra la llamada (CU-11).	

Identificador	UC-11
Nombre	Cierre de llamada
Descripción	Finaliza la llamada cortando así la interacción entre el paciente y el agente inteligente.
Actores	Paciente, agente inteligente.
Precondiciones	El agente inteligente brindó la información requerida al cliente. El paciente no desea continuar con el flujo normal de eventos.
Postcondiciones	La llamada finaliza.
Flujo normal de eventos	
1. El agente inteligente culminó con el flujo normal de eventos. 2. El agente inteligente termina la llamada con el cliente. 3. El agente inteligente es deshabilitado por parte del sistema.	
Flujos alternos y excepciones	
1A. El paciente desea interrumpir el flujo normal de eventos. 2A. El paciente termina la llamada. 3A. El agente inteligente es deshabilitado por parte del sistema.	

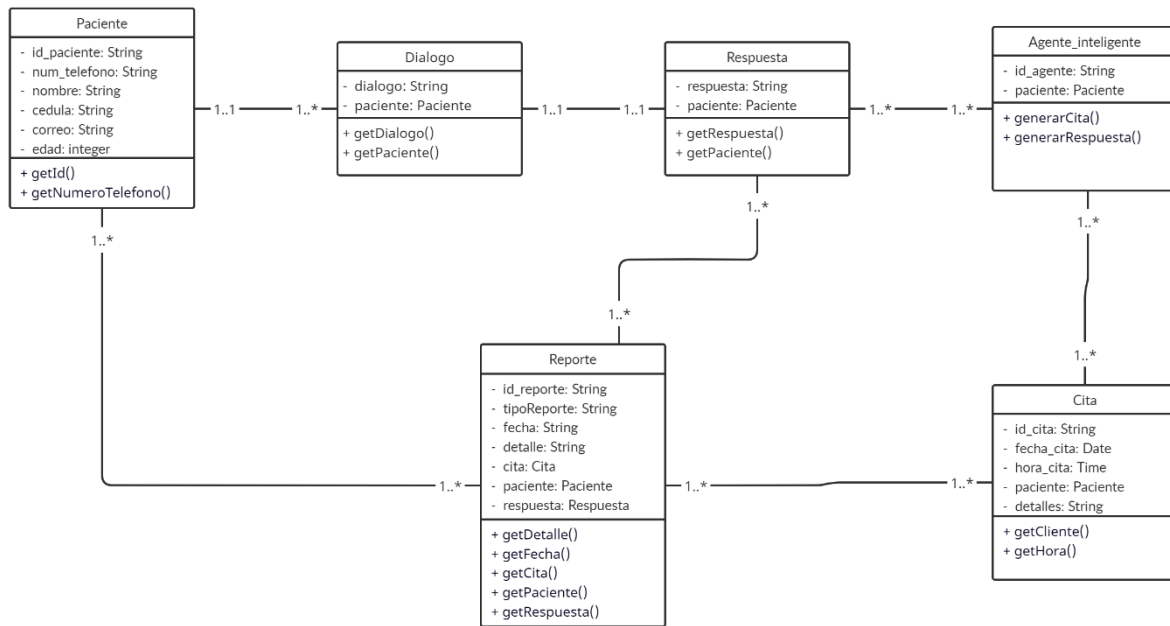
Identificador	UC-12
Nombre	Generar reportes
Descripción	Permite al administrador del sistema la generación de reportes: reportes de usuarios que ingresan al centro de llamadas, reporte de citas generadas por el sistema y reportes de los medicamentos recetados a los diferentes pacientes.
Actores	Paciente, agente inteligente.
Precondiciones	Debe existir registros en la fecha indicada para la generación de los reportes.

Postcondiciones	Se entregará un reporte con la información solicitada.
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción “Generar reportes”. 2. El sistema despliega los diferentes reportes que se pueden generar. 3. El administrador selecciona la opción “Generar reportes de los usuarios que han ingresado al centro de llamadas”. 4. El administrador selecciona el rango de la fecha del reporte. 5. El administrador hace click en el botón “generar reporte”. 6. El sistema genera y despliega una vista con el reporte seleccionado. 	
Flujos alternos y excepciones	
<ol style="list-style-type: none"> 3A. El administrador selecciona la opción “Generar reporte de las citas generadas por el sistema”. 4A. El administrador selecciona el rango de la fecha del reporte. 5A. El administrador hace click en el botón “generar reporte”. 6A. El sistema genera y despliega una vista con el reporte seleccionado. 3B. El administrador selecciona la opción “Generar reporte de los medicamentos recetados por el sistema”. 4B. El administrador selecciona el rango de la fecha del reporte. 5B. El administrador hace click en el botón “generar reporte”. 6B. El sistema genera y despliega una vista con el reporte seleccionado. 	

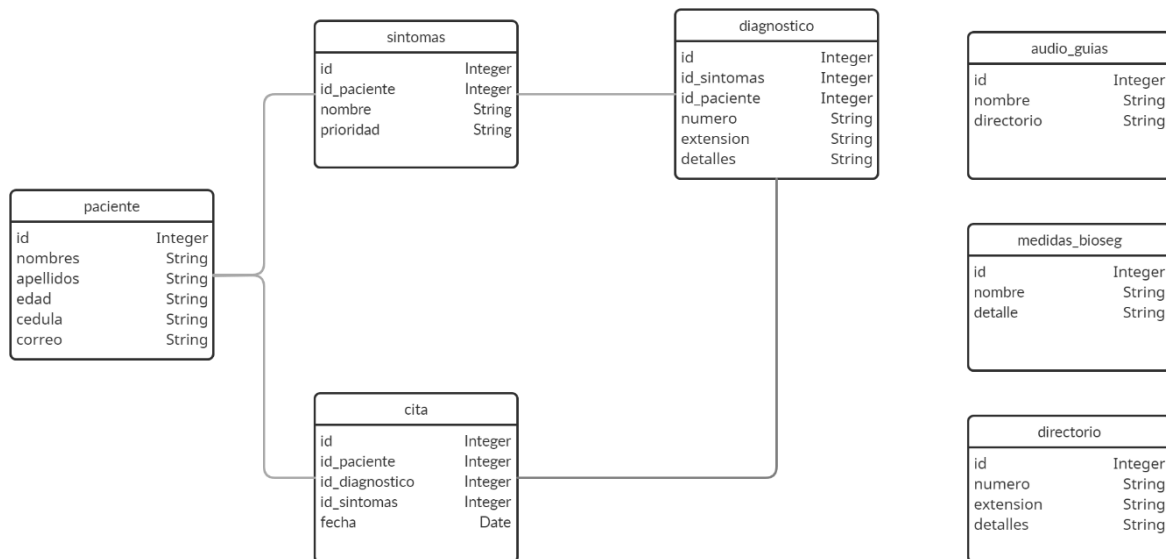
3.2. Vista Lógica

En esta vista se representa la funcionalidad que el sistema proporcionara a los usuarios finales.

3.2.1. Diagrama de clases



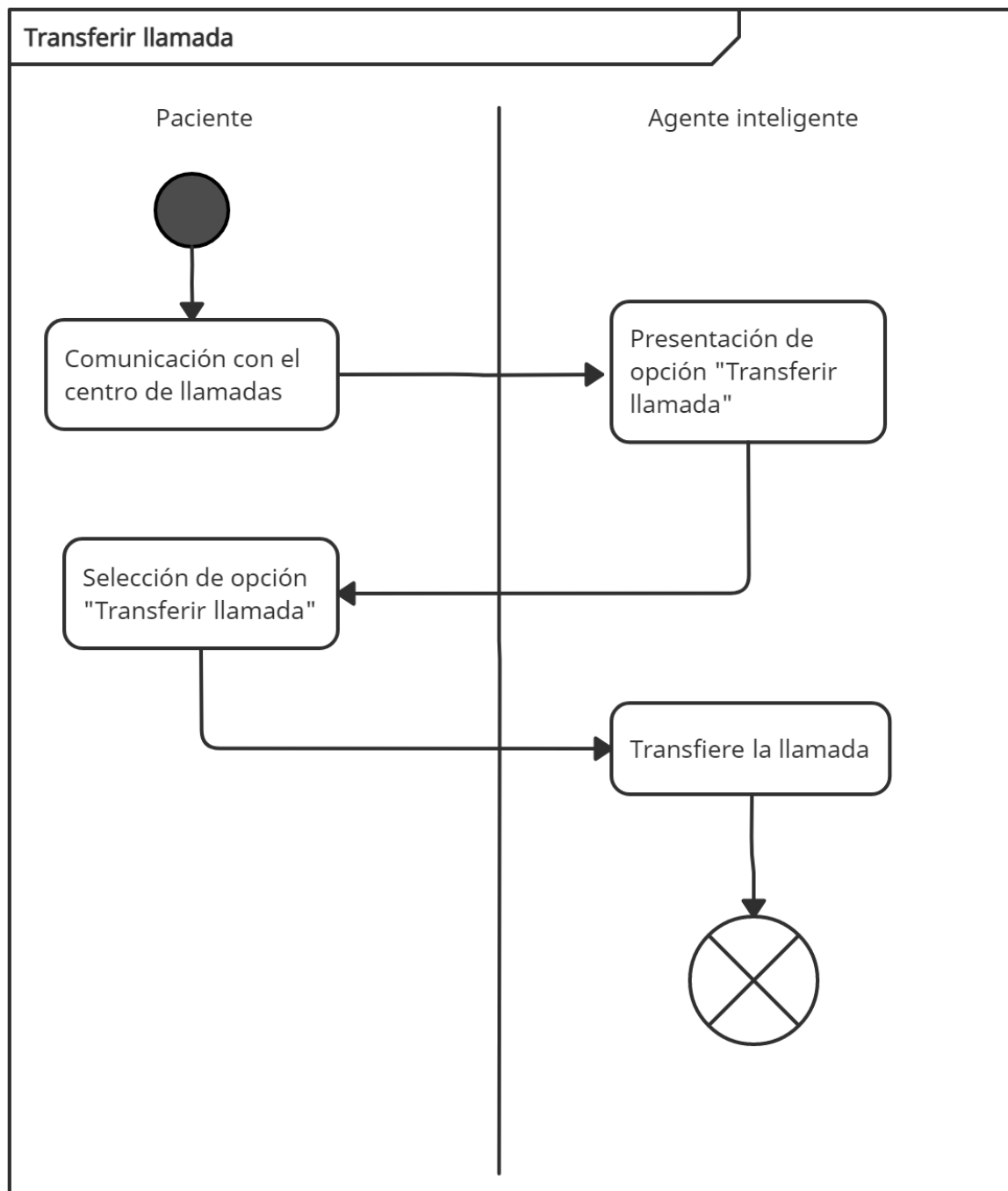
3.2.2. Base de datos



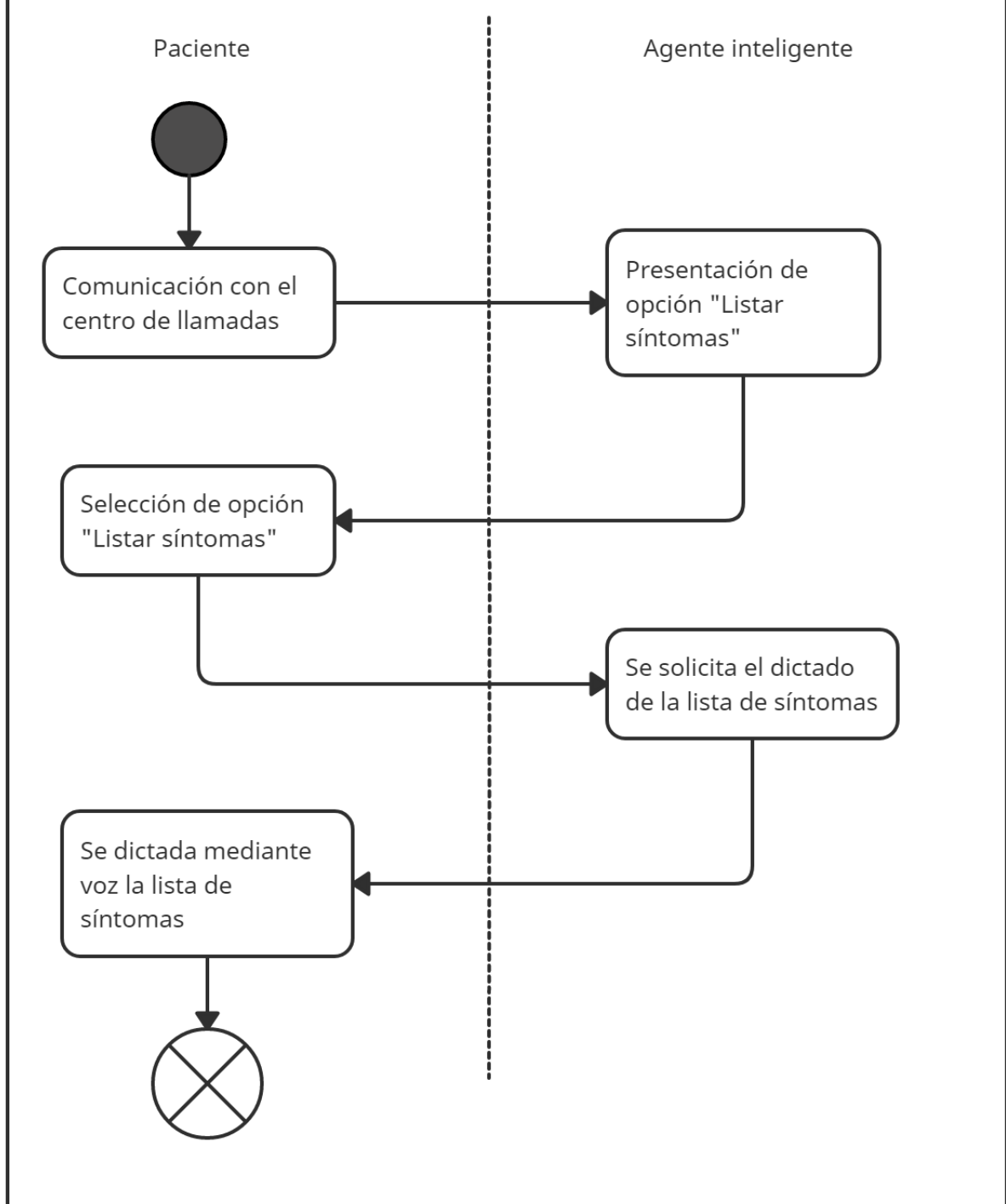
3.3. Vista de Procesos

En esta vista se muestra los procesos que hay en el sistema y la forma en la que se comunican estos procesos.

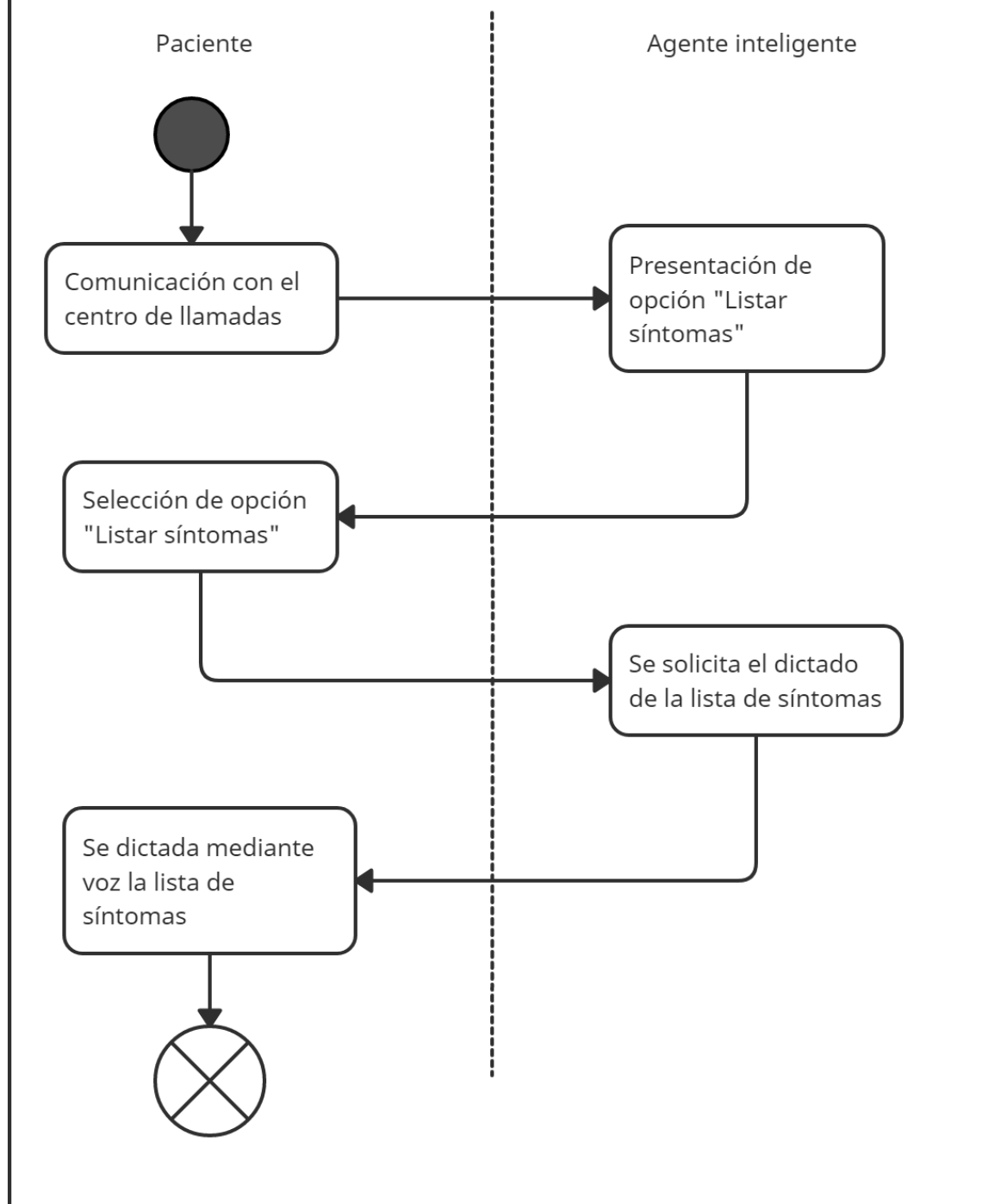
3.3.1. Diagrama de Actividad



Listar síntomas



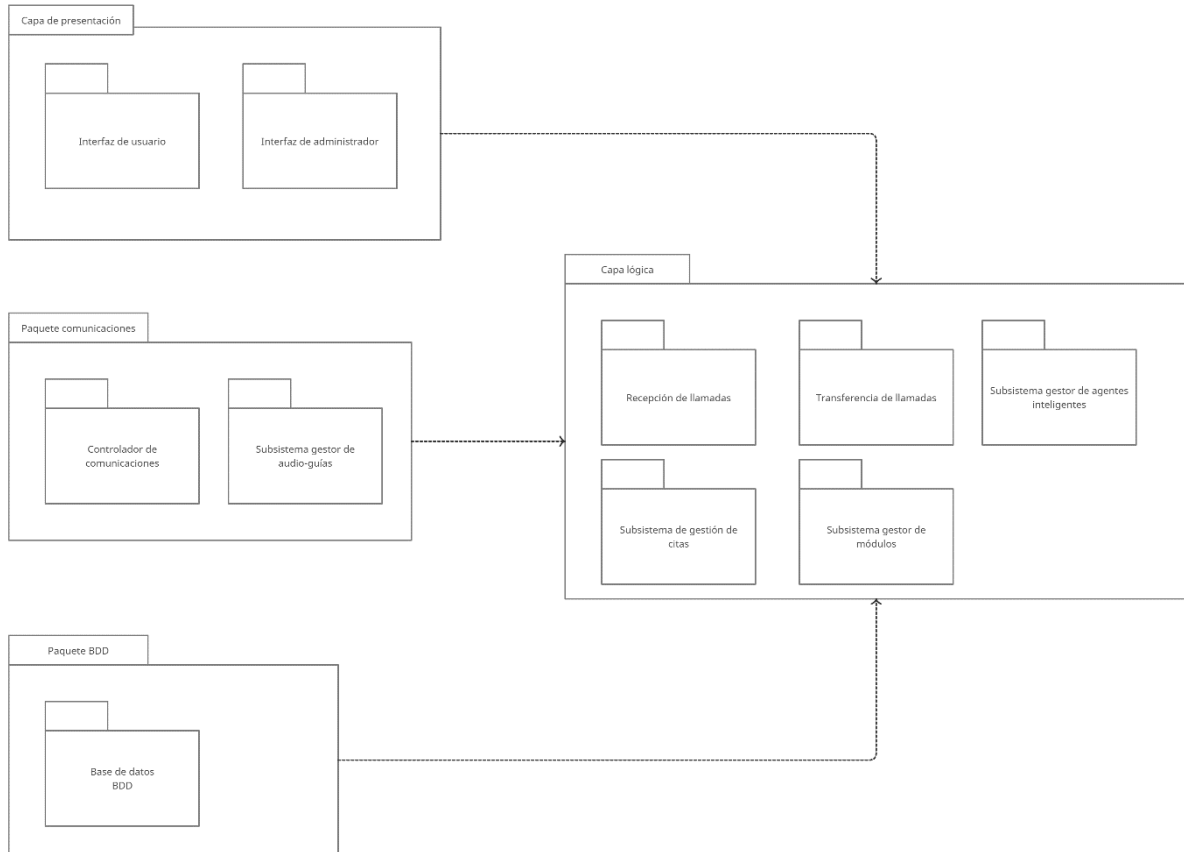
Clasificación de síntomas



3.4. Vista de Despliegue

En esta vista se muestra cómo está dividido el sistema software en componentes y las dependencias que hay entre esos componentes.

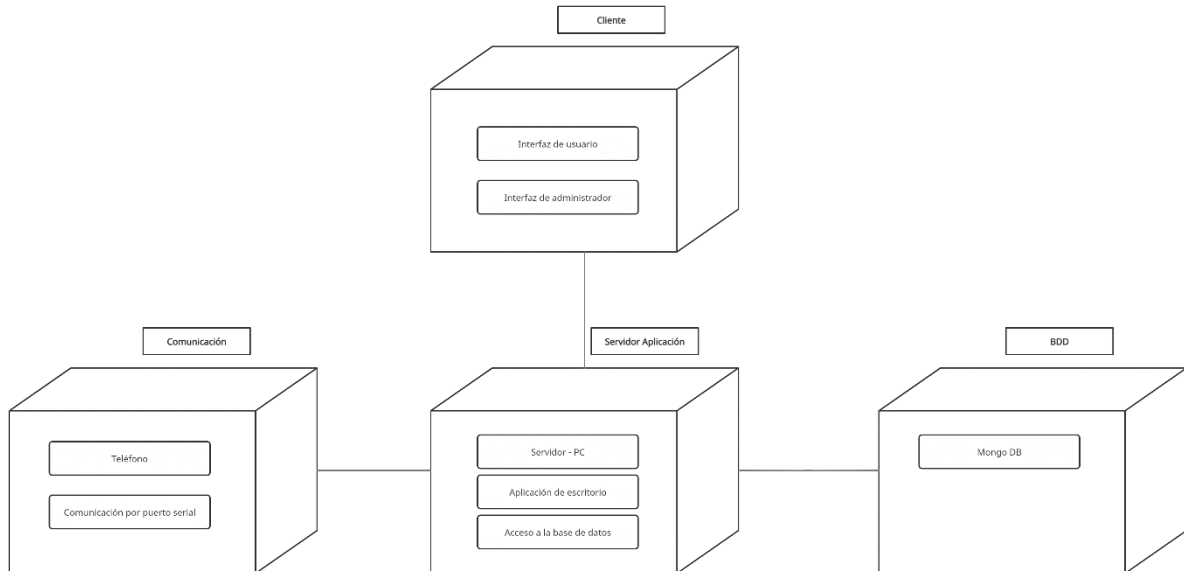
3.4.1. Diagrama de paquetes



3.5. Vista de Física

En esta vista se muestra todos los componentes físicos del sistema, así como todas las conexiones físicas entre esos componentes que conforman la solución.

3.5.1. Diagrama de despliegue



4. Esquema general

