

UNIVERSIDAD ICESI

DEPARTAMENTO DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIONES
PROGRAMA INGENIERÍA TELEMÁTICA
FACULTAD DE INGENIERÍA



Documento Final Proyecto de Grado

Sistema móvil para la gestión de la solicitud y el traslado a una institución médica utilizando una ambulancia en la ciudad de Cali

INTEGRANTES

JUAN FELIPE GÓMEZ M.

LUZ ÁNGELA SARRIA Q.

DIRIGIDA POR

ÁLVARO PACHÓN

SANTIAGO DE CALI, MAYO DE 2014

RESUMEN

De manera sucinta, este documento tiene como finalidad presentar la solución al problema del intercambio de información entre las ambulancias y las clínicas del país, con un enfoque en la ciudad de Cali y limitado a un escenario de prueba e investigación de una clínica en la ciudad.

Los objetivos del proyecto se centran en la elaboración de un conjunto de entregables durante el desarrollo de la solución. Dichos entregables consideran: en primer lugar, una investigación del funcionamiento y la operación de un sistema de urgencias en una clínica, y por supuesto, la comunicación de éste con las ambulancias. En segundo lugar, se propone el diseño de un servicio móvil que permita el envío de información relacionada con los signos vitales de un paciente mientras es transportado en una ambulancia desde un sitio remoto a un centro médico. En tercer lugar, el diseño e implementación de una aplicación que brinde una solución sencilla y eficiente para la comunicación y el envío de información entre las ambulancias y las clínicas teniendo en cuenta la solicitud de la atención por parte de un usuario. Por último, se propone la evaluación de la solución propuesta a través de la ejecución de una prueba piloto en un escenario limitado que permita valorar su funcionamiento.

Entre los motivantes destacados que impulsan la realización del proyecto se encuentran: la oportunidad de mejorar la transmisión de datos relevantes de las condiciones del paciente, la recepción y la clasificación de su urgencia de una manera más rápida utilizando el tiempo de traslado a la clínica. Este tipo de comunicación permite que las interacciones entre las ambulancias y las clínicas sean medibles, es decir, al guardar registros de estas interacciones, se pueden realizar auditorías y correcciones pertinentes con solo revisar la base de datos de dichos registros. Además, el uso de Tecnologías de Información y Comunicación es valorado en un país en desarrollo como lo es Colombia, ya que la aplicación contribuye en el campo de la telemedicina.

Se espera que este proyecto, una vez terminado, sea adaptable a las diferentes necesidades y variables que requieran ser transmitidas. Sin embargo, la mayor contribución con este proyecto, radica en dos campos alejados como lo son la medicina y la ingeniería telemática, los cuales, al integrarse ayudan a la comunidad, pacientes en este caso, quienes son los beneficiados con esta aplicación. En conclusión, la medicina busca contribuir a mejorar la calidad de vida de las personas con problemas de salud, y la ingeniería telemática basa sus contribuciones tecnológicas en asegurar la disponibilidad de la información en el momento y en el lugar que se necesite, es decir, la información precisa en el momento indicado ayuda a la labor del médico, por lo tanto la sinergia que permite la aplicación contribuirá a mejorar la calidad de vida de los pacientes transportados en ambulancia a las clínicas.

ABSTRACT

Briefly this document is to present the solution to a problem posed regarding communications between ambulances and clinics in the country, with a focus on the Cali city and limited to a test scenario and research clinic in the city.

Project objectives focus on the realization of deliverables made during the development of the solution. These deliverables, primarily consists of an investigation about the functioning and operation of an emergency system in the clinic and of course, communicating this with the ambulances. The second is based on the design of a mobile service that allows sending information related to vital signs of a patient while being transported from an ambulance to a medical facility. The third consists of the design and implementation of an application that provides a simple and efficient solution for communication and transmission of information between ambulances and clinics considering request for attention from a user. Finally, evaluation of a pilot test of the solution in a limited scenario allowing to assess its operation arise.

The motivation that drives the project is located in: the opportunity to improve the transmission of relevant conditions of the patient data, the reception and classification of urgency of a faster way, using the time transfer from an ambulance to a clinical. Additionally, setting this type of communication, allows interactions between ambulances and clinics are measurable, that is to say, to keep records of these interactions can contribute to audits and corrections with just check the database of such records. Furthermore, the use of Information and Communication Technologies is valued in a developing country like Colombia, because the application contributes to the field of telemedicine.

It is expected that this project once completed is adaptable to the different needs and variables that need to be transmitted. However, the greatest contribution to achieve with this project lies in two distant fields such as medicine and telematic engineering give results that help the community, patients in this case, who are the beneficiaries of this application. In conclusion medicine seeks to improve the quality of life of people with health problems, and telematics engineering bases its technological contributions to ensure the availability of information at the time and place it is needed, that is to say, the accurate information at the right time helps the physician's work, therefore the synergy that allows the application will help improve the quality of life of patients transported by ambulance to the clinics.

TABLA DE CONTENIDO

Resumen.....	2
Abstract	3
Lista de Figuras	6
Lista de Tablas	8
Lista de acrónimos.....	9
Glosario de Términos	10
1. Motivación y Antecedentes.....	12
2. Descripción del problema.....	14
3. Objetivos del proyecto	15
3.1. Objetivo General	15
3.2. Objetivos Específicos	15
4. Marco Teórico y Estado del arte.....	16
4.1. Contexto.....	16
4.1.1. Modelo de atención de ambulancias en Instituciones de Salud.	16
4.1.2 Tipos de Ambulancias	17
4.2. Marco Legal.....	18
4.3. Factores Determinantes.....	19
4.4. Estado del Arte.....	20
5. Contribución del proyecto de grado.....	23
5.1. Contribución relacionada con el objeto del proyecto.....	23
5.2. Contribución relacionada con el proceso de formación profesional	23
5.3. Contribución relacionada con el proceso de formación personal	23
5.4. Resultados que se esperan obtener	24
6. Metodología	25
6.1. Esquema de trabajo	25
6.2. Fases de desarrollo del proyecto	25
7. Desarrollo del Proyecto	28
7.1. Documentación de Análisis.....	28
7.2. Caracterización del Problema	28
7.3. Especificación Funcional	31
7.4. Requerimientos Funcionales.....	35
7.4.1 Front End	35
7.4.2 Requerimientos No Funcionales.....	39
7.4.3 Requerimientos del Back-End.....	41
7.4.4 Requerimientos Back-End generales: notificaciones Emergentes	42

7.5.	Documentación de Diseño	42
7.6.	Especificación del Diseño Funcional	42
7.6.1	Procesos según el Tipo de Emergencia.....	42
7.6.2	Modelos Gráficos Interfaz de Usuario (Mock-Ups)	44
7.6.3	Diagrama de navegación	47
7.7.	Especificación del Diseño Técnico.....	49
7.7.1	Modelo de Conceptos.....	49
7.7.2	Diseño Base de Datos	51
7.7.3	Modelo de Mensajes	57
7.7.4	Diagrama de clases	60
7.8.	Desarrollo del Sistema	64
7.9.	Componente de Front End.....	65
7.9.1	Software de Desarrollado Instalado	65
7.9.2	Preparación de ambiente de desarrollo	65
7.10.	Implementación Aplicaciones Móviles	66
7.10.1	Aplicación Usuario.....	67
7.10.2	Aplicación Paramédico	72
7.10.3	Aplicación Vigilante	79
7.10.4	Aplicación Médico	83
7.11.	Componente de Back End	87
7.11.1	Software de Desarrollado Instalado.....	88
7.11.2	Preparación de ambiente de desarrollo.....	90
7.12.	Implementación Base de Datos	96
7.13.	Implementación de Servicios Web.	100
7.13.1	Servicio Web Almacenamiento	100
7.13.2	Servicio Web Usuario.	110
7.13.3	Servicio Web Paramédico	113
7.13.4	Servicio Web Institución Médica.....	116
8.	Reportes de Prueba	119
8.1	Ambiente de Pruebas	120
8.2	Casos de prueba	121
	Trabajo a Futuro	121
	CONCLUSIONES	122
	Referencias Bibliográficas.....	123
	Agradecimientos.....	126

LISTA DE FIGURAS

Figura 1. Metodología SCRUM	26
Figura 2. Diagrama Solicitud por paciente.	30
Figura 3. Diagrama Solicitud por EPS o IPS.....	31
Figura 4. Proceso Emergencia Vital	33
Figura 5. Proceso Emergencia No Vital.....	34
Figura 6. Proceso traslado interhospitalario	35
Figura 7. Mock-up App Usuario-paciente	44
Figura 8. Mock-Up App Ambulancia	45
Figura 9. Mock-Up App Médico.....	45
Figura 10. Mock-Up App Vigilante	46
Figura 11. Mock-Up Aplicación web Admisiones	46
Figura 12. Mock-Up Aplicación web Administradores del Sistema	46
Figura 13. Diagrama de Navegación	48
Figura 14. Diagrama de Relación entre conceptos.	50
Figura 15. Protocolo de comunicación	58
Figura 16. Diagrama de comunicación usuario final y webservices	59
Figura 17. Modelo cliente/servidor	60
Figura 18 Diagrama General componentes SIEF	64
Figura 19. Diagrama de flujo Android Studio	66
Figura 20. Reconocimiento de marca	69
Figura 21. Cargando posición actual.....	69
Figura 22. Posición actual encontrada.	69
Figura 23. Búsqueda de ambulancia.....	70
Figura 24. Ruta entre ambulancia y ubicación inicial	70
Figura 25. Instituciones médicas cercanas	71
Figura 26. Calificar	71
Figura 27. Calificación /saltar/enviar	71
Figura 28. Cargue inicial, reconocimiento de marca	73
Figura 29. Pantalla Login	74
Figura 30. Pantalla de Bienvenida	74
Figura 31. Cambio de estado.	75
Figura 32. Momento de emergencia o solicitud de servicio	75
Figura 33. Pantalla de nuevo traslado	76
Figura 34. Pantalla ingreso de signos vitales	76
Figura 35. Seleccionar una institución cercana	77
Figura 36. Pantalla ingreso de información del paciente	77
Figura 37. Traslado en curso.....	78
Figura 38. Envío de mensaje en el camino	78
Figura 39. Traslado interhospitalario	79
Figura 40. Lista de servicios atendidos.	79
Figura 41. Reconocimiento de marca.....	81
Figura 42. Pantalla de Login.....	81
Figura 43. Menú de bienvenida.....	82

Figura 44. Mensaje de emergencia en camino.....	82
Figura 45. Pantalla información del traslado.....	83
Figura 46. Lista de traslados a confirmar.....	83
Figura 47. Pantalla inicial de marca.....	85
Figura 48. Pantalla de login.....	85
Figura 49. Mensaje nueva emergencia.....	86
Figura 50. Pantalla información del traslado.....	86
Figura 51. Pantalla datos de peciente en camino.....	87
Figura 52. Pantalla de lista de servicios atendidos.....	87
Figura 53. Componentes de arquitectura.....	88
Figura 54. Instalación MySQL.....	90
Figura 55. Instalación JDK.....	91
Figura 56. Instalación Development fir.....	92
Figura 57. Instalación Eclipse.....	93
Figura 58. INSTALACIÓN APP ENGINE.....	95
Figura 59. Modelo Relacional Base Datos.....	96
Figura 60 Configuración prueba piloto.....	99
Figura 61. Manejo de Procesos.....	100
Figura 62. modelo general almacenamiento en servicio web.....	101
Figura 63. ACLARACIÓN DE PROCESOS.....	103
Figura 64. EJEMPLO CÓDIGO BÚSQUEDA.....	104
Figura 65. Métodos de ALMACENAMIENTO.....	105
Figura 66. Indicador WebService.....	106
Figura 67. ingreso manual de de usuarios al WebService.....	106
Figura 68. Explicación Adaptador Servicio Web.....	107
Figura 69. Estructura SOAP.....	108
Figura 70. Comunicación SOAP.....	109
Figura 71. Muestra General Proceso.....	111
Figura 72. Funcionalidad Métodos Usuarios.....	112
Figura 73. Proceso General Servicio Web Paramédico.....	114
Figura 74. Métodos Servicio Web Paramédico.....	115
Figura 75. Proceso General Servicio Web IPS.....	117
Figura 76. Métodos Servicio Web Institución Médica.....	118
Figura 77. Etapas de un ambiente de pruebas.....	120

LISTA DE TABLAS

Tabla 1. Base de datos institución	52
Tabla 2. Base de datos empleado.....	52
Tabla 3. Base de dato de los roles	52
Tabla 4. Base de datos ambulancias.....	53
Tabla 5. Base de datos de ubicaciones	53
Tabla 6. Base de datos paciente	54
Tabla 7. Base de datos de enfermedades.....	54
Tabla 8. Base de datos Paciente/Enfermedad	54
Tabla 9. Base de datos de procedimientos.....	55
Tabla 10. Base de datos paciente/procedimiento.....	55
Tabla 11. Base de datos EPS	55
Tabla 12. Base de datos traslado.....	56
Tabla 13. Tabla Comparativa	57
Tabla 14. Versiones Eclipse.....	94
Tabla 17. Muestra tabla institución.....	97
Tabla 18. tabla base de datos SIEF	102

LISTA DE ACRÓNIMOS

TIC	Tecnologías de la información y la comunicación
OMS	Organización Mundial de la Salud
EPS	Entidad Promotora de Salud
NTC	Norma Técnica Colombiana
TAB	Transporte Asistencial Básico
TAM	Transporte Asistencial Medicalizado
NEDOCS	National Emergency Department Overcrowding Study Scale
SOAT	Seguro Obligatorio de Accidentes de Tránsito
WLAN	Wireless Local Area Network
ePCR	electronic Patient Care Reporting
PC	Personal Computer
VPN	Virtual Private Network
EMS	Emergency Medical Service
SAMU	Servicio de Atención Médica de Urgencias
IMC	Índice de Masa Corporal
IPS	Institución Prestadora de Salud
EPS	Entidad Prestadora de Salud
CAD	Centro Automático de Despacho
SIEF	Sistema Integrado de Emergencias Funcional
JDK	Java Development Kit
SDK	Software Development Kit
SOAP	Simple Object Access Protocol
GPS	Global Positioning System

GLOSARIO DE TÉRMINOS

A

Avantel: Empresa privada Colombiana que provee servicios de Trunking Digital de comunicación inmediata con tecnología de red digital e integrada (avantel, 2014).

D

Documentar: Este término hace referencia a realizar una evaluación detallada del paciente en el momento de la atención de una emergencia. Se incluye información como el nombre, edad, cuadro clínico, procedimientos realizados, alergias, EPS y signos vitales.

E

Escala de Glasgow: La escala de Glasgow es una valoración del nivel de conciencia en la cual se evalúan tres criterios de observación clínica: la respuesta ocular, la respuesta verbal y la respuesta motora. Fue creada en 1974 por Bryan Jennett y Graham Teasdale y es una herramienta ampliamente utilizada para valorar de manera objetiva el estado de conciencia para las víctimas de traumatismos craneoencefálicos (Salud Madrid, 2011).

G

Gestograma: Un gestograma es una herramienta utilizada para calcular cuántas semanas de embarazo tiene actualmente un paciente, además, de presentar en forma clara la fecha de parto y otros datos sobre el desarrollo del bebé. Se trata de un calendario circular con fechas en el círculo más exterior y otras cifras en el círculo interior, que también recibe el nombre de disco gestacional (SemanaaSemana.com, 2008).

H

Hospital: Es un centro de salud destinado a la prevención, diagnóstico, y si es posible, cura de enfermedades físicas y/o psíquicas, incluyendo la realización de prácticas quirúrgicas, comprendiendo también la atención odontológica (deconceptos.com, 2014).

I

Índice de Masa Corporal: El Índice de masa corporal (IMC), estima el peso ideal de una persona en función de su tamaño y peso, este es válido para cualquier adulto entre los 18 y 65 años. El IMC también se conoce como el índice de Quételet, debido a que fue desarrollado por el científico belga Jacques Quételet. La Organización Mundial de la Salud (OMS) ha definido este índice de masa corporal como el estándar

para la evaluación de los riesgos asociados con el exceso de peso en adultos (indicemasacorporal.org, 2007).

T

TIC: Tecnologías de la información y las comunicaciones.

Triage: es un término de origen francés, actualmente aceptado por toda la comunidad médica mundial, significa la clasificación de pacientes según su estado de salud. Genéricamente consiste en un conjunto de procedimientos sencillos, rápidos y repetitivos, efectuados sobre cada una de las víctimas que en ese momento demanda asistencia y que orientan sobre sus posibilidades de supervivencia como consecuencia del proceso que le afecta (Álvarez Leiva & Macías Seda, 2001).

Trunking Digital: Son redes confiables de radio tecnología de alto rendimiento que aprovechan el espectro radioeléctrico de manera automática para establecer una comunicación segura (Novarum, 2007).

V

Video Streaming: El video streaming es la tecnología que permite la retransmisión de archivos multimedia (audio o video) a través de Internet. La diferencia principal con los métodos tradicionales, es que para poder ver un video no hace falta descargar el archivo completo en el computador. Mediante el video streaming, el servidor, comienza a enviarnos fragmentos del archivo en el mismo momento que lo solicitamos y a una velocidad acorde con la capacidad de nuestra conexión a Internet (Universidad de Alicante, 2014).

1. MOTIVACIÓN Y ANTECEDENTES

Las telecomunicaciones han sido utilizadas de diversas formas en el área de la salud. Los telégrafos, teléfonos, la radio y enlaces satelitales se han usado desde su creación para solicitar, guiar o apoyar la asistencia médica, sobretudo en lugares lejanos y donde las condiciones geográficas hacen difícil la atención presencial requerida. La utilización de las telecomunicaciones en la medicina permitió acuñar el término de telemedicina. Luego, en la década de los noventa, el uso intensivo y empresarial de la Internet y el advenimiento de las Tecnologías de la Información y la Comunicación (TIC) permitieron desarrollar la telemedicina que, según la definición de la Organización Mundial de la Salud (OMS), puede ser entendida como: *“El suministro de servicios de atención sanitaria, en los que la distancia constituye un factor crítico, por profesionales que apelan a las TICs con objeto de intercambiar datos para hacer diagnósticos, preconizar tratamientos y prevenir enfermedades y heridas, así como para la formación permanente de los profesionales de atención de la salud y en actividades de investigación y evaluación, con el fin de mejorar la salud de las personas y de las comunidades en que viven”*. A partir de esta definición, y conforme con los desarrollos tecnológicos, la telemedicina se encuentra relacionada con la posibilidad de desarrollar/diseñar cualquier proceso en la atención sanitaria utilizando redes de comunicación y sistemas interoperables de información. Este término también es conocido a nivel mundial como E-Health, e Salud, Salud digital o tele salud (Sanidad, 2007; Monteagudo, 2001).

En Colombia, propiamente en la ciudad de Cali, surge la necesidad de agilizar la recepción y atención de pacientes en la Unidad de Urgencias Médicas, pues dadas las condiciones de accidentalidad y violencia en la ciudad, estas unidades permanecen generalmente congestionadas, y ante este hecho, se desea poder coordinar a todos los actores involucrados para mejorar la atención de los pacientes y mejorar las estadísticas de prestación de servicio. Por lo tanto, se genera una oportunidad de emplear el tiempo de traslado de un paciente en una ambulancia al centro médico, produciendo y transmitiendo información al lugar de recepción adelantando la clasificación de tipo de la atención requerida (TRIAGE) (Velandia Escobar, 2011).

Bajo este esquema la realización de aplicaciones, sistemas, redes y/o dispositivos que aporten avances en telemedicina son relevantes, máxime en nuestro país, donde éste campo se encuentra en una fase inicial de aplicación. El continuo crecimiento en aspectos tecnológicos representa un impacto positivo para el país, y ser partícipes de la solución aportando una aplicación de telemedicina, se convierte en un gran motivante.

En Colombia, Telemed es el centro encargado del despliegue de la Telemedicina, desde el año 2003 brinda información sobre la organización y los diferentes eventos realizados, conferencias, simposios y seminarios (Telemedicina Colombia, 2013). Entre las diferentes memorias publicadas destacamos las ventajas y beneficios que los países como Brasil, México, Francia han obtenido con implementaciones de telemedicina. Por ejemplo, México en la participación de las conferencias del año 2012 afirmó un ahorro con la implementación de sistemas de telemedicina de \$31,900 pesos mexicanos mensuales, los cuales irían en aumento conforme al avance en el proceso de implementación (Pacheco López, 2012).

Según lo dicho en Colombia Telemed, la aplicación que se propone desarrollar se encuentra enmarcada dentro del concepto de M-medicine (Movilidad en la atención médica y de salud, utilizando las Tecnologías de la Información y las Comunicaciones). En consecuencia, es necesario tener un conocimiento en las TICs, específicamente en el área de la telemática, en la conceptualización, diseño, construcción y despliegue de aplicaciones móviles. Aunque se han implementado soluciones parecidas en otros países, para el caso colombiano una aplicación como esta aún no existe, por lo que se debe analizar la atención de urgencias que presenta el sistema de salud en nuestro país, para tener un impacto mayor con la aplicación y lograr los objetivos planteados.

2. DESCRIPCIÓN DEL PROBLEMA

En la actualidad, específicamente en la ciudad de Cali, una ambulancia es usada principalmente como un medio que permite transportar un paciente desde el lugar (el sitio donde tiene lugar la urgencia o una institución médica) hasta la unidad de urgencias de un centro médico, como un hospital o una clínica, en el menor tiempo posible, en este tiempo, la prioridad es mantener estable y con vida al paciente mientras se llega al destino final. En ocasiones, cuando el paciente llega a la unidad de urgencias de la institución de salud, no se cuenta con información básica, ni un diagnóstico de su condición actual. Por lo tanto, resulta necesario efectuar una valoración de su estado (clasificación del TRIAGE). Una vez se conoce el nivel de gravedad del paciente, se empiezan a realizar los procedimientos necesarios para atenderlo, salvarle la vida y dejarlo fuera de peligro.

Con lo anteriormente mencionado, se plantea el problema que no existe en la ciudad de Cali un servicio que gestione de forma eficiente e integrada la solicitud, el traslado y la coordinación de actividades entre el paciente, la ambulancia y la institución médica. Tres factores que influyen significativamente en el problema, el primero es la deficiente comunicación utilizada por las partes -ambulancia e institución médica- la cual, actualmente, se da a través de un canal de voz, debido a esto no es posible tener una comunicación concurrente y una conexión estable, además, de no se tiene un registro de la información que se ha intercambiado; el segundo factor es la limitada tecnología existente en la mayoría de ambulancias de la ciudad, ya que no existe un dispositivo capaz de enviar datos, de las condiciones del paciente a bordo, a la unidad de urgencias; el tercer elemento importante que debe ser considerado es el tiempo de espera de un paciente para la atención de la urgencia por parte de la ambulancia, se puede dar el caso en que la ambulancia se demore un tiempo considerable en atender la emergencia y la vida de la persona se ponga en riesgo.

3. OBJETIVOS DEL PROYECTO

3.1. OBJETIVO GENERAL

Desarrollar y evaluar un servicio móvil que permita gestionar la solicitud, el traslado y la interacción entre el paciente, la empresa de ambulancias y la institución médica de forma eficiente e integrada.

3.2. OBJETIVOS ESPECÍFICOS

- Caracterizar el funcionamiento de un sistema de urgencias en una clínica de la ciudad de Cali, identificando los requerimientos para el servicio móvil propuesto.
- Diseñar la arquitectura del servicio móvil que permita la gestión eficiente e integrada de la solicitud, el traslado y la interacción entre el paciente, la empresa de ambulancias y la institución médica.
- Implementar el servicio móvil que permite la gestión eficiente e integrada de la solicitud, el traslado y la interacción entre el paciente, la empresa de ambulancias y la institución médica.
- Diseñar e implementar una aplicación, con una interfaz gráfica de usuario sencilla y eficiente, que permita a un usuario con smart phone solicitar la atención de una ambulancia para una emergencia.
- Evaluar el servicio móvil utilizando un escenario limitado que permita valorar su funcionamiento y el cumplimiento de las métricas de desempeño establecidas.

4. MARCO TEÓRICO Y ESTADO DEL ARTE

4.1. CONTEXTO

La Telemedicina es la provisión de servicios de salud a distancia, en los componentes de promoción, prevención, diagnóstico, tratamiento o rehabilitación por profesionales de la salud que utilizan Tecnologías de la Información y Comunicación, que les permiten intercambiar datos con el propósito de facilitar el acceso de la población a los servicios que presentan limitación de oferta, acceso a los servicio o de ambos en su área geográfica ("Resolución," 2006).

Históricamente, el desarrollo de la Telemedicina ha evolucionado en diferentes etapas, el creciente nivel de complejidad en la tecnología desde la década de 1960, donde fue conocida por primera vez el término, hasta ahora donde se evidencia una revolución en el campo de las telecomunicaciones (Cuenca, 2011). Abreviando, la base sobre la que se apoya cualquier sistema de telemedicina es la infraestructura de comunicación, teniendo en cuenta que los requerimientos para los sistema de comunicación de telemedicina están impuestos por la medicina, quien es la que clasifica el tipo y caudal de información que se debe transferir para obtener un resultado apropiado al contexto que se desarrolle la tecnología.

En pocas palabras, la telemedicina posibilita la optimización de los servicios de atención en salud, que involucren una disminución de tiempo y costos.

4.1.1. MODELO DE ATENCIÓN DE AMBULANCIAS EN INSTITUCIONES DE SALUD.

La atención en Urgencias de las diferentes Instituciones de Salud se rigen bajo disposiciones comunes entre las que se destacan: La Atención Inicial de Urgencias “Modificación circular Externa No. 049 de 2008 concerniente a las Instituciones Prestadoras de Servicios de Salud deberán establecer un protocolo para la atención de urgencias médicas, encaminado a fijar prioridades así como a exigir efectividad, calidad y rapidez en la atención de sus usuarios en especial a los niños, niñas y adolescentes”. Para estos efectos deberán instruir a todo su personal administrativo y médico. [Inciso adicionado por la Circular Externa 052 de 2008]. Por consiguiente, las urgencias se clasifican en cuatro categorías según el TRIAGE; la primera categoría es usada para los casos de Emergencia Extrema o Emergencia Vital, la etiqueta en esta categoría es la Roja. La segunda categoría es usada para lo urgente, el color de esta categoría es la Amarilla. La tercera categoría clasifica lo no urgente y el color usado es Verde, por último, la cuarta categoría en la que se ubican los Fallecidos, la etiqueta usada aquí es de color Gris.

Todos los usuarios que ingresen a la unidad de urgencias en un medio diferente a una ambulancia deben esperar a que le realicen la clasificación del TRIAGE, y posteriormente ser atendidos según la prioridad y prelación de otros enfermos (Álvarez Leiva & Macías Seda, 2001). Por otra parte, si el paciente es ingresado en ambulancia, el proceso es diferente, porque la clasificación de la urgencia (TRIAGE) la realiza el médico encargado de forma inmediata, y según la clasificación, si se decide que el paciente ingresa a la clínica, el paramédico realiza una entrega del paciente directamente al médico de urgencias. Si la clasificación de urgencia es roja debe ser atendido de forma inmediata, de lo contrario el centro médico solicita autorización a la Entidad Promotora de Salud (EPS) del paciente para su atención en dicha entidad. La variación de la clasificación se encuentra en los procedimientos y protocolos de atención dispuestos por las diferentes instituciones de salud. Sin embargo, si el paciente presenta una urgencia vital, es trasladado a la unidad médica más cercana sin importar si existe cupo en esta; pero si el paciente a trasladar no presenta una extrema urgencia y se encuentra en capacidad de esperar, los paramédicos llaman vía telefónica a las diferentes entidades prestadoras de salud en busca de cupo para el paciente según la EPS del mismo. Este proceso puede tardar, dado el caso que en un tiempo aproximado de dos horas no se consiga cupo en alguna de las clínicas con convenio de la EPS, los paramédicos deben llevar al paciente a su EPS. Una vez trasladado el paciente, el procedimiento de verificación de estado y condiciones del mismo se realizan en el momento de la entrega, esto se hace de forma verbal entre el paramédico y el médico de turno en urgencias. Quedando pendiente el proceso de registro digital del ingreso a la clínica del paciente y la historia clínica realizada en la ambulancia, ya que dicha documentación se entrega en papel en admisiones. Una vez completados estos procesos, termina el ingreso de un paciente en una entidad de salud.

4.1.2 TIPOS DE AMBULANCIAS

En primer lugar, es necesario especificar la definición de Ambulancia terrestre: *“Vehículo de emergencia autorizado para transitar con prioridad de acuerdo con la condición del paciente y acondicionada de manera especial y exclusiva para el transporte de pacientes, con recursos humanos y técnicos calificados para la atención y beneficio de aquellos. Es un vehículo automotor y como tal está regido por las normas nacionales pertinentes, expedidas por el Ministerio de Transporte, sin poseer ningún beneficio adicional al asignado en esta definición. Por tanto, no tiene ninguna prerrogativa diferente a la de ser un vehículo con circulación prioritaria”* (Novarum, 2007). Respecto a las generalidades de estos vehículos en nuestro país se deben cumplir las reglamentaciones dispuestas por los Ministerios de la Protección Social, del Medio Ambiente y de Transporte a través de los decretos, artículos y resoluciones vigentes a la fecha de la publicación de los términos y relacionadas para este tipo de vehículos, así como lo

dispuesto en el Código Nacional de Tránsito Terrestre, las Normas Técnicas Colombianas del ICONTEC: NTC 2170, la NTC 3729 tercera actualización y las demás normas que las adicionen, modifiquen o sustituyan. Las normativas van desde detalles técnico-mecánicos hasta la dotación de implementos y tripulación. En cuanto al diseño las ambulancias, constan de dos áreas principales denominadas compartimento del conductor y compartimento del paciente, las cuales deben ser independientes total o parcialmente, garantizando como mínimo la comunicación visual y auditiva entre sí, a través de acceso directo, ventana o puerta (no plegables), de fácil limpieza y desinfección. Lo siguiente a estos requerimientos es la separación de tipo de ambulancias que varía la dotación de implementos según el país. Empero Colombia clasifica las ambulancias de acuerdo a su ámbito de servicio:

a) Ambulancia de transporte asistencial básico (TAB): Unidad móvil destinada al transporte y/o asistencia de pacientes cuyo estado real o potencial no precisan cuidado asistencial médico durante la atención y el transporte (Comité técnico, 2011).

b) Ambulancias de transporte asistencial medicalizado (TAM): Unidad móvil destinada al transporte y/o asistencia de pacientes cuyo estado potencial y/o real es de riesgo y requieren equipamiento, material y personal médico durante la atención y el transporte (Comité técnico, 2011).

Las ambulancias TAB se distinguen de las TAM por el equipamiento de dispositivos, insumos médicos y por la tripulación a bordo, siguiendo el orden, el recurso humano debe estar conformado por el conductor y un auxiliar de enfermería o de ambulancias. A su vez las TAM deben contar con enfermera jefe o auxiliar de enfermería, paramédico conductor y un médico, los cuales deberán acreditar un entrenamiento mínimo de 200 horas, teórico práctico, para el manejo del paciente crítico en ambulancia, en institución pública o privada aprobada por el gobierno (Novarum, 2007; "Resolución," 2006; Cuenca, 2011).

4.2. MARCO LEGAL

El marco normativo por el cual se sustentan la organización de los componentes relacionados en el proyecto, tales como las ambulancias y las unidades de urgencia de los centros médicos es la siguiente:

1. Artículo 64, Ley 769 agosto 06 de 2002, Ministerio de Transportes. Cesión de paso en la vía a vehículos de emergencia y chequeo del vehículo para mantenerlo en condiciones óptimas.
2. El personal que tripula ambulancias terrestres debe cumplir los requerimientos mínimos mencionados en la Ley y establecidos en la Resolución 1439 (Manual de estándares y procedimientos. Ministerio de Protección Social anexo 1-30).

3. Ley 1438 de 2011 en su artículo 67 del Gobierno Nacional a través del Ministerio de Salud y Protección Social, aborda el tema de la coordinación y articulación de todos los actores que intervienen en la atención integral de las emergencias médicas.
4. Escala de NEDOCS utilizada para medir la congestión en Urgencias.
5. Reglamentan los servicios de urgencias, Decreto 1761 de 1990, Reglamenta parcialmente los servicios de urgencias, Decreto 412 de 1992.
6. Ley 1419, con la cual el Congreso estableció los lineamientos para la telesalud en Colombia.

4.3. FACTORES DETERMINANTES

La situación de congestión a nivel de los servicios de urgencias es un problema que ocurre a nivel mundial y aún no tiene un marco de definición claro, para el cual se ha tratado de plantear soluciones pensadas de acuerdo al diseño y funcionamiento de los servicios de urgencias de cada país. Actualmente, en Colombia se ha decidido utilizar la escala de NEDOCS (National Emergency Department Overcrowding Study Scale) para realizar la medición de la congestión de usuarios en las salas de urgencias, dicha escala es un instrumento de cinco preguntas que fue desarrollado en 8 servicios de urgencias de instituciones académicas (Castro Canoa et al, 2010) y luego validado contra la medición del sobrecupo desde el punto de vista de la carga asistencial del personal médico y de enfermería ("Decreto," 2007; Comité técnico, 2011). Entre los cinco ítems que determina ésta escala se encuentra el mayor tiempo de espera para los pacientes de urgencias (en horas) y el tiempo en la sala del último paciente acostado en cama en el servicio de emergencias. Según las mediciones, en términos generales, en nuestro país el personal asistencial y los usuarios terminan por acostumbrarse a “convivir con la congestión” e identifican, como momentos de congestión, sólo aquellos donde se ven sometidos a situaciones de estrés o de presión externa incrementada por parte de las ambulancias (Secretaría Salud, 2014). Esto se debe a que el proceso de entrega de un paciente que ingresa en ambulancia al equipo de urgencias del hospital es mano a mano con algún médico del hospital. A partir de ese momento, el médico es quien asume la responsabilidad del enfermo. La entrega del paciente en el hospital conlleva una completa información tanto verbal como escrita de los puntos a destacar en la actuación extra hospitalaria por parte del paramédico. Se realiza un breve historial clínico, con los datos obtenidos, incluyendo la hora, lugar y mecanismo del accidente, la valoración clínica realizada y las medidas de reanimación vital que han sido precisas (Dirección Sanidad, 2010). Lo que se define como ideal es hacer esta entrega en una zona de recepción de emergencias, prevista en cada centro. La cual ahora es deficiente y engorrosa. Para el paramédico el traslado y transferencia finaliza en ese punto de entrega de documentos, pues solo le resta recuperar la operatividad en el menor tiempo posible, reponiendo el material

empleado y acondicionando el interior del vehículo para cualquier nueva situación de emergencia.

El factor tiempo es clave en los acontecimientos que contemplan una emergencia. Existe controversia entre el sistema privado y el Seguro obligatorio de Accidentes de Tránsito (SOAT), el primero que tiene tendencia a tomarse el tiempo necesario para estabilizar al paciente antes de su traslado al hospital, y el SOAT basado en "cargar y correr", que realiza una evacuación rápida, depositando mayor confianza en el ámbito hospitalario. La diferencia estriba en los tipos de ambulancias utilizados, equipos y personal a bordo (médico, paramédico).

Sin lugar a dudas, la implementación de un servicio que permite usar el tiempo de traslado para el envío de información y datos de regulación de ingreso de pacientes a las entidades sería un factor que juega a favor de las posibilidades de supervivencia de los que necesitan tratamiento hospitalario y quienes requieren intervención especializada inmediata, lo que precisa de preparación para la atención por parte de las clínicas.

4.4. ESTADO DEL ARTE

El problema planteado en la segunda sección de este documento, claramente, se presenta en diferentes sociedades alrededor del mundo, problema que países con una inversión fuerte en investigación y desarrollo han logrado dar solución de diversas maneras con aplicaciones y sistemas completos, ligados al campo de la telemedicina, que permiten optimizar el tiempo de traslado de un paciente, mediante el intercambio de información crítica (como signos vitales, procedimientos realizados, clasificación TRIAGE) al centro médico; gracias a estos sistemas es posible la ejecución de diferentes procesos al interior de la ambulancia que ayudan, indudablemente, a salvar la vida de un paciente. Además, por parte del centro de urgencias, al cual es llevado el paciente, hay una gran ventaja a la hora de conocer la situación actual del mismo, debido a que se tiene todo listo para tratarlo, es decir, una vez llega el paciente a la institución el personal encargado de atenderlo se encuentra preparado con todos los equipos necesarios listos, para realizar un tratamiento que saque de peligro a dicho paciente. Este tipo de aplicaciones encuentran su mayor potencial en zonas de difícil acceso, ya que los tiempos de traslado son mucho mayores a los presentados en un entorno urbano, de esta manera aprovechar dicho tiempo se convierte en una variable crucial para conservar la vida de un paciente. Por lo tanto, es necesario conocer las soluciones planteadas en diferentes naciones para lograr comprender porque no es posible adaptarlas a la situación vivida actualmente en Colombia específicamente, en la ciudad de Cali.

En primer lugar, se tiene una solución implementada en un país desarrollado, Estados Unidos, donde ya existe un sistema encargado de transmitir los signos vitales de un paciente en tiempo real desde una ambulancia hacia un Hospital Central de Traumas (Tovar López). Dicha aplicación

fue desarrollada en la Universidad de Maryland, en Baltimore, en conjunto con una empresa influyente en el área de la salud; esta aplicación permite investigar la factibilidad y el modo práctico de transmitir datos de signos vitales, imágenes y video en tiempo real de la atención de pacientes desde la ambulancia al centro médico utilizando una comunicación inalámbrica, ya sea vía Celular o tecnologías WLAN (Wireless Local Area Network) (Tovar López). Este procedimiento no es posible aplicarlo en el país debido a la poca tecnología presente en las ambulancias, además, de ser un sistema costoso, en el cual es muy difícil realizar una inversión monetaria.

Otra aplicación existente en Estados Unidos se denomina AmbuPad, la cual es un producto resultante del trabajo de un grupo de desarrolladores denominado WebMedicPro, este es un ePCR (Electronic Patient Care Reporting) (EMS1, 2014), robusto y sencillo de utilizar por el personal presente en una ambulancia. Dicho ePCR presenta una alta adaptabilidad debido a que el AmbuPad es independiente del sistema operativo, por ende es posible ejecutarlo en cualquier clase de dispositivo, tales como: iPad, PC, MAC, Smart Phones (con Android, Windows phone o iOS) y Kindle; pero es ampliamente utilizada en iPads por las ventajas técnicas -hardware de mejor calidad- presentadas por esta clase de dispositivos. Actualmente el AmbuPad es soportado por la red de Verizon y la transmisión de información se realiza de manera segura, cifrando cada dato, usando una VPN (Virtual private Network). Por último, el AmbuPad no es considerado un software, sino un instrumento de Emergency Medical Service (EMS). Por esta razón es visto en el mismo nivel de importancia que un desfibrilador o un electrocardiograma; esto se debe a la función que realiza el AmbuPad dentro de una ambulancia, como rastrear y monitorear el cuidado de los pacientes. Adicionalmente, esta solución presenta una gran ventaja con respecto a otras, ya que, no requiere tiempo para ser puesta en práctica, tampoco requiere inversiones adicionales para mantenimiento, es por esto que resulta ser una solución muy rentable para una entidad de salud (webMedicpro, 2013). Un instrumento como el AmbuPad es lo más parecido al servicio móvil que se tiene pensado desarrollar, pero con un nivel de complejidad mucho mayor que requiere más tiempo de trabajo al estipulado en este proyecto de grado; aunque puede resolver el problema planteado, ciertas regulaciones colombianas frenarían la implementación de una solución como el AmbuPad.

En el caso de un país más cercano, como lo es Chile, también se han implementado varias soluciones al problema planteado, una de ellas desarrollada la ITMS Chile S.A. La solución propuesta se denomina Tele UTI, la cual consiste en una caja electrónica que mide los parámetros vitales de un paciente, dicha caja se le puede adosar a la persona sin ningún inconveniente, además, la información recopilada es transmitida vía BlackBerry a la central hospitalaria donde es analizada por un especialista que dará instrucciones precisas para el tratamiento del enfermo. Esta práctica empezó a ser implementada en el año 2011 y

actualmente es usada también en Brasil, donde 320 ambulancias del SAMU (Servicio de Atención Médica de Urgencia) de los 114 servicios de salud poseen soporte de avanzada. Otro procedimiento interesante y que indudablemente contribuye a evitar el desconocimiento, de la situación del paciente, en las salas de urgencia es llamada Tele-Consulta; esta plataforma puede operar en lugares remotos donde no hay especialistas, por ende, médico y paciente pueden relacionarse a través de una consola que permite un flujo de información pertinente para ayudar a resolver los problemas de salud; esta aplicación, resulta ser una consulta a distancia con nuestro médico por video-streaming (ITMS Chile, 2011). Ambas soluciones presentan las mismas dificultades para ser implementadas en Colombia, ya que son soluciones costosas que sólo resuelven una parte del problema planteado, igual continúa existiendo una comunicación deficiente entre ambulancia y centro de urgencias.

En el caso colombiano, este tipo de aplicaciones no es posible encontrarlas por lo mencionado anteriormente, pero si existe un centro encargado del desarrollo para la telemedicina, como se mencionó anteriormente en los antecedentes. Aunque este centro trabaja activamente en la implementación de la telemedicina en nuestro país, la realidad es otra; en Colombia no se tienen políticas o regulaciones que inciten a las entidades de salud a invertir en dispositivos tecnológicos que ayuden al mejoramiento del sistema de salud, y mucho menos una inversión en investigación en telemedicina, por lo que su avance es un poco lento, por ende tan solo 7.500 colombianos se han visto beneficiados (enticconfio.gov, 2012). Debido a estas pobres regulaciones, las EPS invierten en dispositivos para dar un valor agregado a su organización y poder ofrecer una mejor calidad de servicio a sus afiliados. Un ejemplo es Coomeva, una EPS que, en la ciudad de Cali, dota a cada uno de sus paramédicos con una Tablet para facilitar los procesos de registro del paciente, esta Tablet tiene instalada 4 App relacionadas con temas de salud, una para la historia clínica del paciente, una para gestogramas, índice de masa corporal y otra para calcular la escala de Glasgow (Peña, 2014).

En conclusión, podemos decir que el problema planteado tiene diversas soluciones implementadas en países como Estados Unidos, Chile y Brasil, mostrando resultados favorables para el sistema de salud, lo cual resulta ser beneficioso para los usuarios, ya que se puede aprovechar mejor el tiempo de traslado en la ambulancia y ganar tiempo valioso para la vida de una persona. Además, se puede decir que es posible adaptar alguna solución al contexto colombiano, como el AmbuPad, tomando como referencia las ambulancias de Coomeva que tienen Tablets con conectividad por medio de la red celular.

5. CONTRIBUCIÓN DEL PROYECTO DE GRADO

5.1. CONTRIBUCIÓN RELACIONADA CON EL OBJETO DEL PROYECTO

1. Aporte en Telemedicina con un servicio móvil que permita mejorar la deficiente comunicación entre una ambulancia y una institución médica.
2. Facilitar el registro de ingreso de pacientes en ambulancias a las clínicas.
3. Contribución al control de la prestación del servicio en salud gracias al registro, en una base de datos, de la información intercambiada (signos vitales, cuadro clínico del paciente, procedimientos realizados, entre otros) para tener un mejor control de la historia clínica del paciente y evitar ciertos inconvenientes legales.

5.2. CONTRIBUCIÓN RELACIONADA CON EL PROCESO DE FORMACIÓN PROFESIONAL

1. Consolidar nuestro conocimientos en el desarrollo de aplicaciones móviles y plataformas en la nube, solucionando un problema evidenciado en la ciudad de Cali y presentan un impacto positivo en la sociedad.
2. Aplicar el ciclo básico de ingeniería: analizar, diseñar, implementar y evaluar una solución en un dominio de problema particular, como lo es el sistema de atención de urgencias.

5.3. CONTRIBUCIÓN RELACIONADA CON EL PROCESO DE FORMACIÓN PERSONAL

Durante el desarrollo del proyecto, a nivel personal cada integrante ha desarrollado y afinado competencias a nivel personal tales como:

1. Trabajo en equipo: Cada miembro tiene establecido de manera individual roles y responsabilidades de trabajo, sin embargo, logramos funcionar como una unidad organizada, gracias a la cohesión expresada en la solidaridad y el sentido de pertenencia del proyecto.
2. Motivación al logro: El interés individual por lograr la finalización con éxito del proyecto y sus diferentes objetivos específicos con una actitud que impulsa a la realización y finalización de la meta propuesta.
3. Solución de Problemas: La identificación del problema, la búsqueda de la solución, encontrar la coherencia y la correspondencia para brindar una solución implementada a problemas reales involucrando conocimientos previos y adquiriendo nuevos en la labor investigativa de la solución.
4. Pensamiento Crítico: visto desde la práctica del proceso del uso de los conocimientos técnicos de la ingeniería telemática para llegar de forma efectiva una posición razonable y justificada sobre los diferentes temas tratados a lo largo del desarrollo del proyecto.

5. Iniciativa: Cada integrante actúa de forma proactiva y con clara orientación al futuro.
6. Comunicación y relaciones interpersonales: Uno de los aspectos más importantes en los que ha contribuido el proyecto es en la comunicación entre los integrantes del grupo, colaboradores, tutor, y profesionales de otras áreas, ya que el intercambio de ideas, experiencias y valores y la transmisión de sentimientos y actitudes se logran a través de una comunicación efectiva.

5.4 RESULTADOS QUE SE ESPERAN OBTENER

Al finalizar este proyecto de grado se desarrollarán los siguientes documentos entregables:

1. Caracterización del Sistema de Urgencias en una clínica de la ciudad de Cali.
2. Documento general de los Requerimientos y la caracterización del algoritmo implementado para la solución.
3. Documento de Arquitectura del servicio móvil propuesto y documentación de usuario y técnica de la aplicación desarrollada.
4. Informe sobre pruebas realizadas y la evaluación final de la efectividad de la propuesta por parte de los usuarios.

Adicionalmente se tendrán 4 aplicaciones móviles que hacen parte del servicio y permiten la interacción de diferentes usuarios con la propuesta de solución:

1. Aplicación móvil que permite a un usuario solicitar una ambulancia cercana para la atención de una emergencia.
2. Aplicación móvil para indicar a la ambulancia los servicios pendientes y permitir el intercambio de información con la institución médica a la cual se dirige, mientras se traslada un paciente.
3. Aplicación móvil para el guarda de seguridad ubicado en la puerta de ingreso a la institución médica, en la cual se puede observar un listado completo de las ambulancias que se dirigen actualmente a dicha institución
4. Aplicación móvil que permite ver el estado actual de un paciente mientras es trasladado en una ambulancia, intercambiando información entre la misma y el médico del TRIAGE en la clínica destino.

6. METODOLOGÍA

6.1. ESQUEMA DE TRABAJO

El esquema de trabajo pactado con el tutor del proyecto de grado cuenta con las siguientes características:

- Veinte horas de trabajo semanal base para el desarrollo del proyecto por parte de los estudiantes, divididas en diez horas cada estudiante.
- Una reunión semanal de control y verificación de avance del proyecto por entre los estudiante y el tutor, cada reunión tendrá una duración de una hora.
- Siempre y cuando lo amerite, reuniones con un asesor, el Dr. Jorge Karim Assis médico de la Clínica Occidente, quien cumple con la función de asesor temático en el proyecto.
- Visitas a las instalaciones de la Clínica de Occidente en el área de Urgencias. De igual forma conocer, visitar y transportarse en los diferentes tipos de ambulancias cuando sea requerido.

6.2 FASES DE DESARROLLO DEL PROYECTO

Para realizar la solución, planteamos 7 fases.

Fases:

1. Documentación y caracterización del servicio móvil: En esta fase se realiza una consulta bibliográfica sobre las propuestas existentes en algunos países, así como la generación de un limitado estudio de caracterización del sistema actual de urgencias en la ciudad de Cali con relación a la comunicación entre este y las ambulancias el cual será consignado en un Short Paper. El desarrollo de esta fase implica el estudio de artículos de investigación, leyes y publicaciones sobre el problema de la interacción entre las emergencias, ambulancias y los hospitales.
2. Análisis de la solución: Siguiendo los entregables de la primera fase se concibe el servicio que da solución levantar los requerimientos en un documento entregable.
3. Definición de modelo de transmisión: Esta fase requiere determinar el medio de transmisión, los protocolos que serán usados, la seguridad requerida en la comunicación y el envío de información. El resultado de esta fase se consignará en el documento general de caracterización del algoritmo implementado para la solución.
4. Diseño de la arquitectura del servicio móvil: Con base en los documentos entregables de las fases anteriores, se realiza el diseño de la propuesta de la comunicación adaptada al contexto de una clínica de la ciudad y una ambulancia. Es preciso para esto realizar la interfaz gráfica de la aplicación de la ambulancia y la clínica. Además de la realización del diagrama de componentes que soportan la arquitectura y el diseño del software del proyecto.

5. Desarrollo de la solución móvil: Se realiza la implementación en código de la aplicación que será manipulada desde la ambulancia. Se entrega un prototipo funcional de la aplicación con su debida documentación.
6. Implementación del servicio móvil en una de las clínicas de la ciudad: Se realiza la instalación, configuración y puesta en operación de la solución móvil para permitir la recepción, almacenamiento y envío de información respectiva de la comunicación. De esta segunda fase de implementación se realiza un prototipo funcional de comunicación y registro de datos, todo código será documentado.
7. Evaluación de desempeño y efectividad de la aplicación: Esta fase se irá desarrollando paralelamente con las fases cinco y seis, realizando las pruebas piloto necesarias para ir corrigiendo el código y lograr al final una implementación funcional y evaluada en un escenario real de la solución propuesta. Durante las pruebas se redactarán informes sobre los resultados obtenidos en cada caso. Por último se pretende realizar una evaluación final a los usuarios beneficiados en el uso de la aplicación en entorno simulado.

Para desarrollar la aplicación móvil se utilizará SCRUM, una metodología ágil y flexible para gestionar el desarrollo de software, en el cual se establecen un conjunto de prácticas y roles que permiten definir el proceso de desarrollo que se ejecutará durante el proyecto con un carácter iterativo e incremental. Ver figura 1:

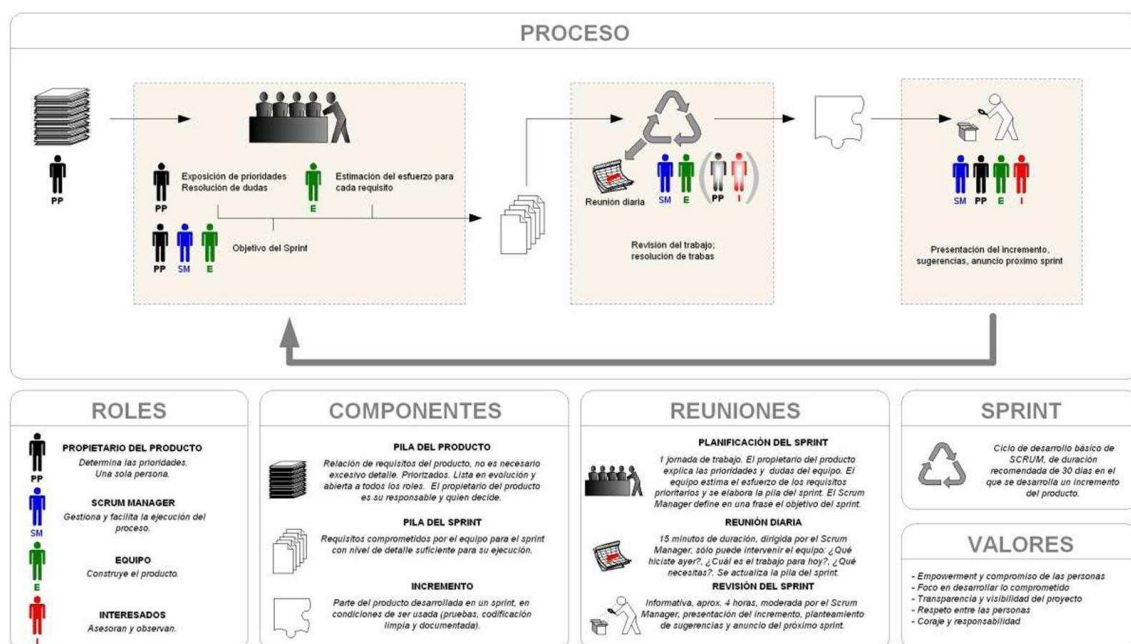


FIGURA 1. METODOLOGÍA SCRUM

En SCRUM, los roles principales son: a) El maestro SCRUM (ScrumMaster), responsable por mantener los procesos y por comportarse como Director del Proyecto; b) El propietario del producto (ProductOwner), que representa las partes interesadas (Stakeholders); y c) El Equipo (Team), que incluye a los desarrolladores. En cada embalaje (sprint), las iteraciones del proceso, el equipo realiza un incremento en la funcionalidad del software. El conjunto de características

que forma parte de cada embalaje viene de la pila del producto (Product Backlog), un conjunto de requerimiento de alto nivel priorizados que definen el trabajo que debe ser ejecutado. Los elementos de la pila del producto que forman parte del embalaje se determinan durante la reunión de planeamiento del embalaje (Sprint Planning). Durante esta reunión el propietario del producto identifica los elementos de la pila del producto que desea completar y los hace de conocimiento del equipo. El equipo determina la cantidad de trabajo que puede comprometerse a completar durante el siguiente embalaje. Durante el embalaje, los requisitos están congelados (Schwaber & Sutherland, 2013).

Esta metodología además ofrece beneficios como: Cumplimiento de expectativas, flexibilidad a cambios, mayor productividad, mayor calidad de software, reducción de riesgos entre otros, por lo tanto, se decide desarrollar la solución de software con dicha metodología.

7 DESARROLLO DEL PROYECTO

En esta sección del documento se describe el desarrollo de cada una de las fases del proyecto descritas previamente. Por cada fase se describe el proceso realizado y los entregables de la fecha que evidencian el resultado obtenido, además, se menciona como dicho entregable ayuda con el cumplimiento de los objetivos específicos.

7.1. DOCUMENTACIÓN DE ANÁLISIS

El sistema de Integrado de Emergencia Funcional (SIEF) integra una solución que contempla 5 frentes de usuario:

1. **Usuario Móvil:** Es la persona que solicita la atención de una emergencia al servicio propuesto, por medio de una aplicación móvil.
2. **Usuario Ambulancia:** Es el usuario responsable de los traslados que se hagan en la ambulancia cumpliendo con la documentación del paciente y estabilizarlo mientras es llevado a una institución médica.
3. **Usuario Médico:** Es el usuario encargado de realizar la atención al paciente en la institución médica. Además, debe estar pendiente de los traslados en camino a la unidad de salud para tener una idea del tratamiento que debe hacerse al paciente.
4. **Usuario Web:** Son los usuarios administradores y se encargan de gestionar la información de cada una de las instituciones presentes en el sistema.

Adicional a esto para su funcionalidad, se requiere dar soporte en base de datos, servicios web (web services) y toda una lógica de coordinación inter aplicaciones.

Los requerimientos de la solución serán divididos en los componentes arquitectónicos: Back End y Front End.

7.2. CARACTERIZACIÓN DEL PROBLEMA

Como se mencionó previamente, para lograr iniciar con un desarrollo claro de la solución propuesta, es necesario hacer una caracterización de la interacción entre las ambulancias y las instituciones médicas, para identificar los procesos que se dan en ambas partes y así, realizar una etapa de análisis más acorde a la situación actual de los sistemas de urgencias de la ciudad de Cali. Para poder culminar esta fase, se realizaron varias visitas a la Clínica de Occidente y junto con una serie de entrevistas realizadas a diferentes personas involucradas en la atención de la emergencia (paramédicos, médicos, vigilantes, personal de admisiones) fue posible tener un documento de caracterización.

Inicialmente, se elaboró una entrevista que pudiera ser aplicada a cualquier persona que estuviera involucrada en los procesos que se definirán posteriormente. Esta entrevista se dividió en tres secciones: la primera, orientada a obtener información de los paramédicos que trabajan en las ambulancias; la segunda, diseñada a los procedimientos de la clínica en el área de urgencias y la tercera, dedicada a tener información técnica que ayudaría en el diseño de la propuesta de solución.

Con las entrevistas realizadas a los paramédicos, se logró identificar el proceso realizado por el personal de la ambulancia a la hora de atender una emergencia. Inicialmente, se identifica el lugar donde se solicita la atención, una vez la ambulancia se encuentra en el sitio, se evalúa la situación y se determina el nivel de urgencia. Si la urgencia es vital, la ambulancia transporta al paciente y se dirige a la Institución Médica más cercana, mientras se encuentra en tránsito, trata de entablar comunicación con el centro médico y documentar el paciente vía voz. Una vez llega la ambulancia con el paciente, la institución debe brindar la atención requerida, sin poner restricciones; en estas situaciones, no se tiene un registro claro de la condición en la que llega el paciente. Por otro lado, si la urgencia es no vital, el paramédico puede documentar el paciente en el lugar de la emergencia y coordinar con la empresa de ambulancias -por ejemplo: cem, EMI, entre otras- la remisión para el cuidado del paciente en una entidad de salud que tenga un convenio con la EPS del paciente; los paramédicos deben esperar a que el traslado sea autorizado. Existe un tercer tipo de traslado denominado traslado interhospitalario, que consiste en realizar un traslado programado para un paciente desde una institución a otra, en este caso, el proceso es más sencillo, ya que no es necesario realizar una solicitud a la institución destino para el ingreso del paciente, ya que está se ha realizado previamente.

Por parte de la clínica, fue posible reconocer los procedimientos realizados en el momento de llegar un paciente. En caso de tener una urgencia que el paramédico considera vital, se dirige hacia la puerta del área de urgencias para informar de la urgencia vital, así, el vigilante busca al médico encargado de realizar el TRIAGE, el cual saldrá y evaluará el nivel de gravedad del paciente; si este médico considera que la urgencia es vital, se ingresará inmediatamente el paciente a la institución, en caso contrario, será necesario realizar el proceso de admisión. Para la situación de una urgencia no vital o traslado de IPS a IPS, el vigilante verifica si el traslado es programado y determina si el paciente puede entrar o si es necesario hacer un proceso adicional. Si el paciente se encuentra en un listado impreso, ingresa sin ningún problema, pero si no está el vigilante se dirige al área de admisiones para verificar si la lista se ha actualizado o si es posible dejar ingresar al paciente para que sea atendido; estas verificaciones se hacen debido a procesos burocráticos que buscan evitar problemas con los pagos de la atención del servicio.

Por último, se tiene las entrevistas técnicas, de las cuales fue posible definir la información puntual requerida por la institución médica - en este caso la Clínica de Occidente- para admitir el cuidado de un paciente y los medios de transmisión idóneos para la transferencia de la información. Además, se utilizó un estándar del área de la salud, el estándar HL7, con el cuál fue posible caracterizar cada una de las enfermedades y procedimientos involucrados en la atención

de cualquier paciente; con esta caracterización se obtuvieron identificadores únicos y descripciones para cada enfermedad y procedimiento.

A continuación, se muestra los diagramas que ejemplifican, de una manera muy sencilla, la forma como se dan los procesos actuales a la hora de atender una emergencia y realizar el traslado de un paciente (ver figura 2 y 3).

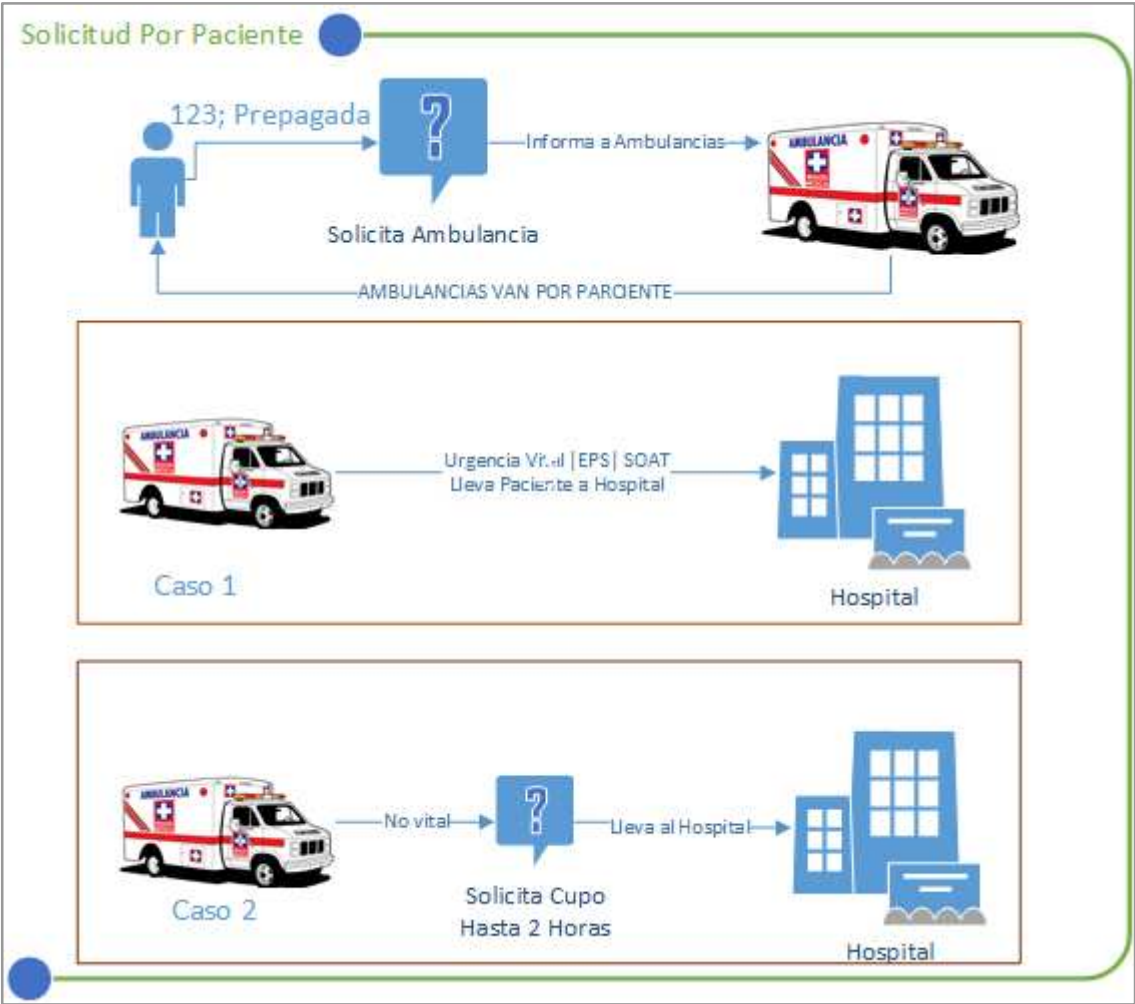


FIGURA 2. DIAGRAMA SOLICITUD POR PACIENTE.

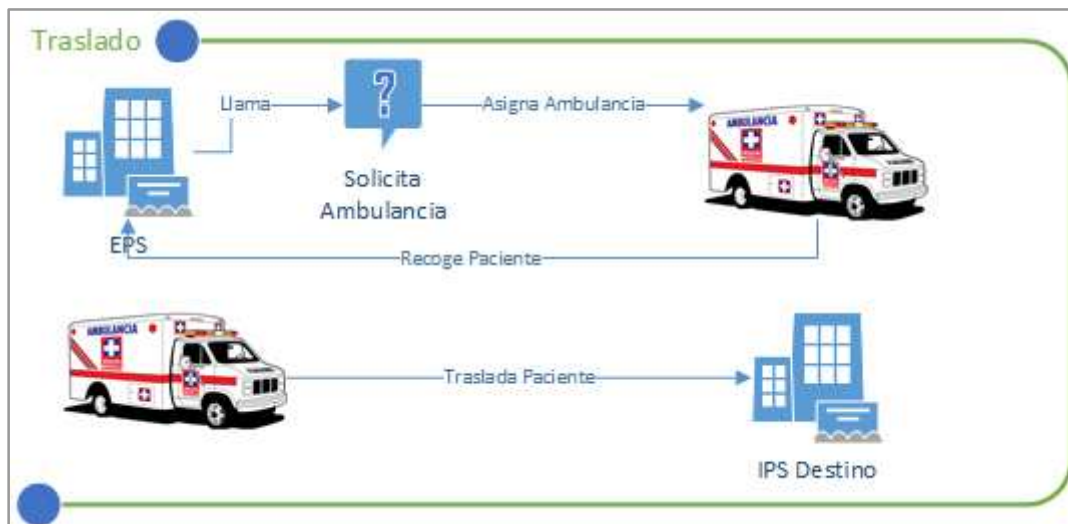


FIGURA 3. DIAGRAMA SOLICITUD POR EPS O IPS

En los anexos “*Caracterización Sistema Urgencias.docx*” y “*ProcesoActual_Urgencias.vsd*” se encuentra descrito de manera detallada cada uno de los procesos que se realizan actualmente entre la ambulancia y la institución médica.

Estos dos entregables ayudan con el cumplimiento del primer objetivo específico, a través de ellos, se logra tener definido claramente los procesos que deben ser considerados en el desarrollo de la solución, caracterizando la situación actual del sistema de atención a urgencias médicas.

7.3. ESPECIFICACIÓN FUNCIONAL

El punto de partida del sistema lo acciona el usuario, el cual desde un dispositivo móvil (Smart Phone) puede acceder a la aplicación. Es importante aclarar que el sistema fue concebido para dispositivos con Android versión 4.0 en adelante.

El acceso a la aplicación permite solicitar un servicio de ambulancia para la ubicación actual o para una dirección específica, además de especificar uno de los tres tipos de emergencia posibles de seleccionar, los cuales son: Hogar, Accidente, Otra.

Una vez solicitado el servicio el sistema se encarga de las consultas a las bases de datos de información, identificando cuales ambulancias se encuentran más cercanas al punto o dirección de solicitud. Dependiendo de la ubicación del evento, a las 3 ambulancias más cercanas se les envía una solicitud de servicio, la cual puede ser aceptada o rechazada. El sistema debe estar presto a recibir la respuesta de las 3 solicitudes; la primera ambulancia que responda afirmativamente será la encargada de iniciar el traslado, a las otras solicitudes se les responderá con una notificación indicando que el servicio ya está siendo atendido por otra ambulancia.

Cuando se determina que ambulancia atenderá la emergencia, se notifica al usuario que realizó la solicitud con la información de ésta, además, se le indica la ruta en el mapa junto con el tiempo estimado de llegada de la misma.

Una vez se confirme la llegada o atención de la ambulancia en el sitio de la solicitud, sea la confirmación por parte del usuario que realiza la solicitud o por parte de la ambulancia que atiende el evento, se guarda la información del servicio y se da inicio al traslado. Para ello, el sistema debe buscar las instituciones médicas más cercanas, las cuales se mostrarán en la aplicación móvil del usuario que solicitó el servicio, posterior a esto el usuario puede terminar la funcionalidad calificando el servicio o saliendo de este sin realizar dicha calificación.

Para la interfaz y/o aplicación disponible en la ambulancia, se inicia el traslado el cual debe indicar el tipo de emergencia encontrar en el lugar: Emergencia Vital, No Vital, Interhospitalaria. Una vez informado esto al sistema se crean tres posibles respuestas:

Emergencia Vital: por Ley, se debe dirigir a la IPS más cercana que dé solución a la emergencia. Por lo tanto, se muestra en el sistema la ruta a la institución médica más cercana al punto de solicitud del servicio. Se presenta la posibilidad de ingresar los signos vitales del paciente.

Si dicha IPS pertenece a SIEF (Sistema Integrado de Emergencias Funcional), se notifica del desplazamiento de la ambulancia y un tiempo estimado de llegada de la misma, avisando el tipo de emergencia y si fue posible en la ambulancia indicar los signos vitales estos llegarán junto con la notificación.

La notificación a la IPS se hace en 3 frentes, Portero, Médico y Admisiones.

Una vez se confirma que el paciente se encuentra en la IPS, desde la aplicación de la ambulancia se debe llenar un breve historial clínico del paciente transportado para poder dar por finalizado el servicio de traslado.

A continuación, se muestra un diagrama que ejemplifica el proceso descrito. Figura 4:

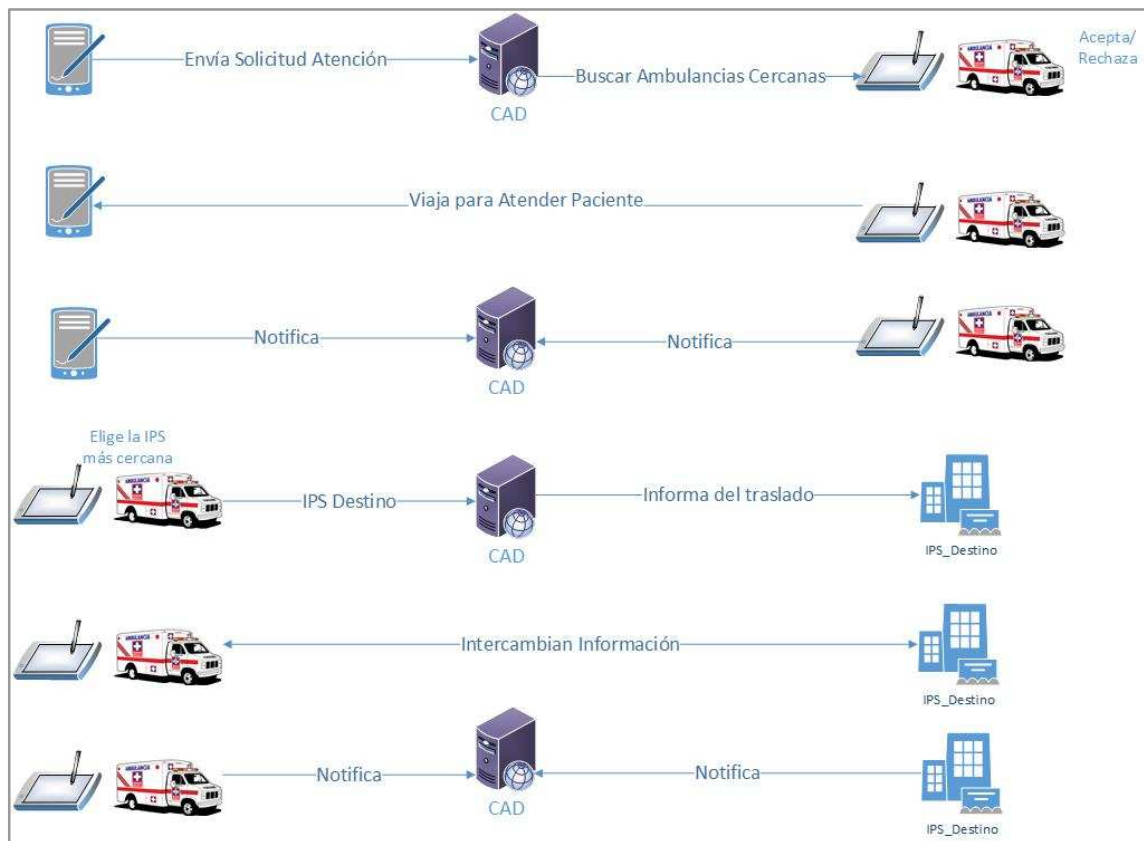


FIGURA 4. PROCESO EMERGENCIA VITAL

Emergencia No Vital: En este caso, el tiempo disponible para realizar un traslado es mayor, por ende, el personal a bordo de la ambulancia puede documentar al paciente suministrando al sistema la información básica del paciente. El sistema brinda 3 opciones de clínica cercanas de las cuales el paramédico puede seleccionar una para dirigirse. Si dicha IPS seleccionada se encuentra en el sistema, será posible notificarle del servicio que se encuentra en camino. En los mismos 3 frentes, portero, médico y admisiones. En la interfaz visual de la aplicación de la ambulancia, se muestra la ruta, el tiempo aproximado de llegada y la posibilidad de enviar un mensaje a la clínica, si ésta tiene el sistema se hará en tiempo real, de lo contrario no sería posible realizar la entrega de dicho mensaje. Ver figura 5.

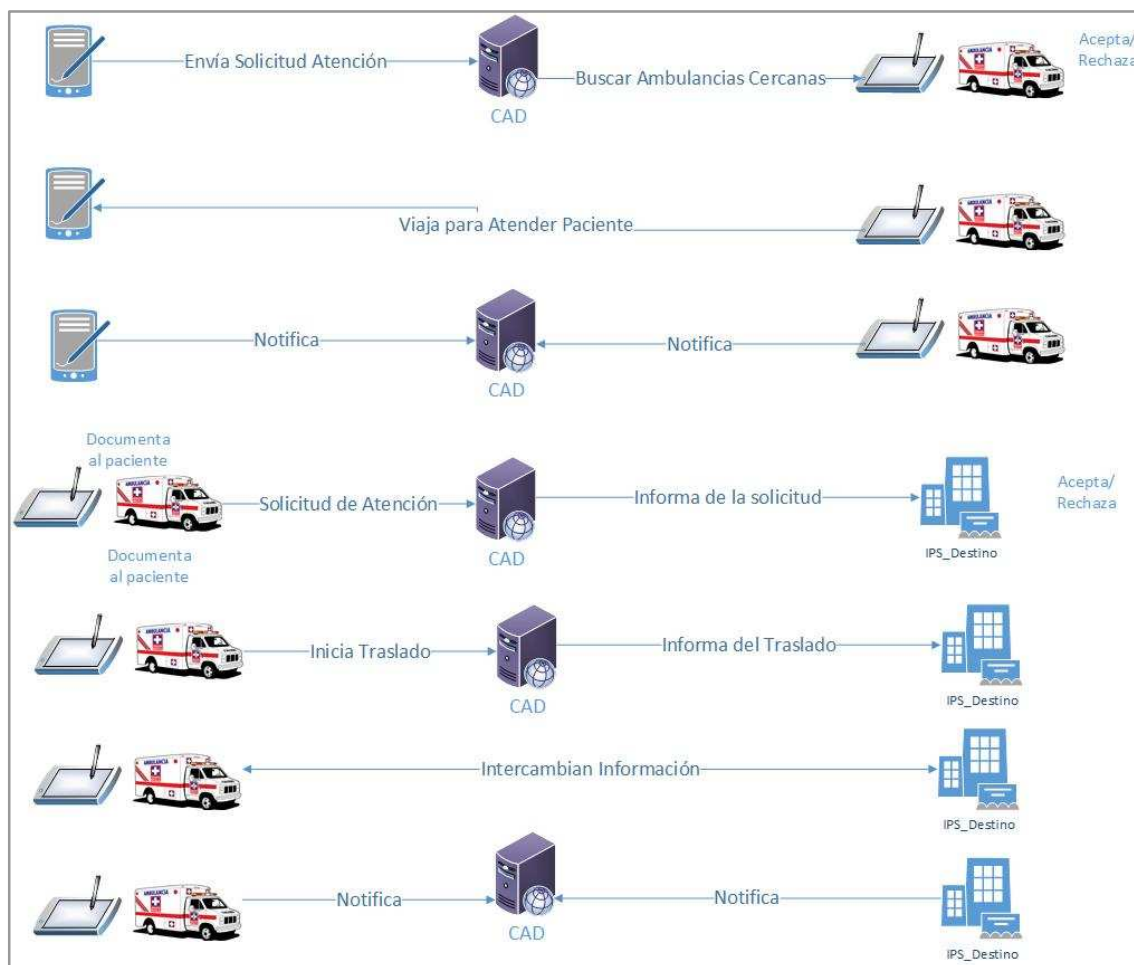


FIGURA 5. PROCESO EMERGENCIA NO VITAL

Interhospitalario: Al llegar la ambulancia al punto de solicitud, se percata que una IPS es la generadora del evento de traslado, por lo tanto, debe indicar el número de autorización del traslado, porque el destino ya ha sido definido en la solicitud. Entonces, en la aplicación de la ambulancia se muestra la ruta al destino, el tiempo estimado y si éste hace parte de SIEF se permitirá el envío de mensajes y/o notificación del traslado en camino.

Las aplicaciones en la IPS para el portero y el médico se usan de manera informativa sobre los traslados próximos a llegar y el historial de los mismos.

La funcionalidad de admisiones se implementa a través del uso de una aplicación Web y funciona tanto para las notificaciones de traslados como para inicio del mismo.

El acceso de los paramédicos, médicos, porteros y personal de admisiones se hace mediante un nombre de usuario que tras la consulta a la base de datos de información de accesos permite o no el ingresar.

El llenado de la base de datos del personal se realiza por parte de los administradores del sistema, por una parte la empresa de ambulancia y por otra la IPS.

La figura 6 ilustra el proceso:

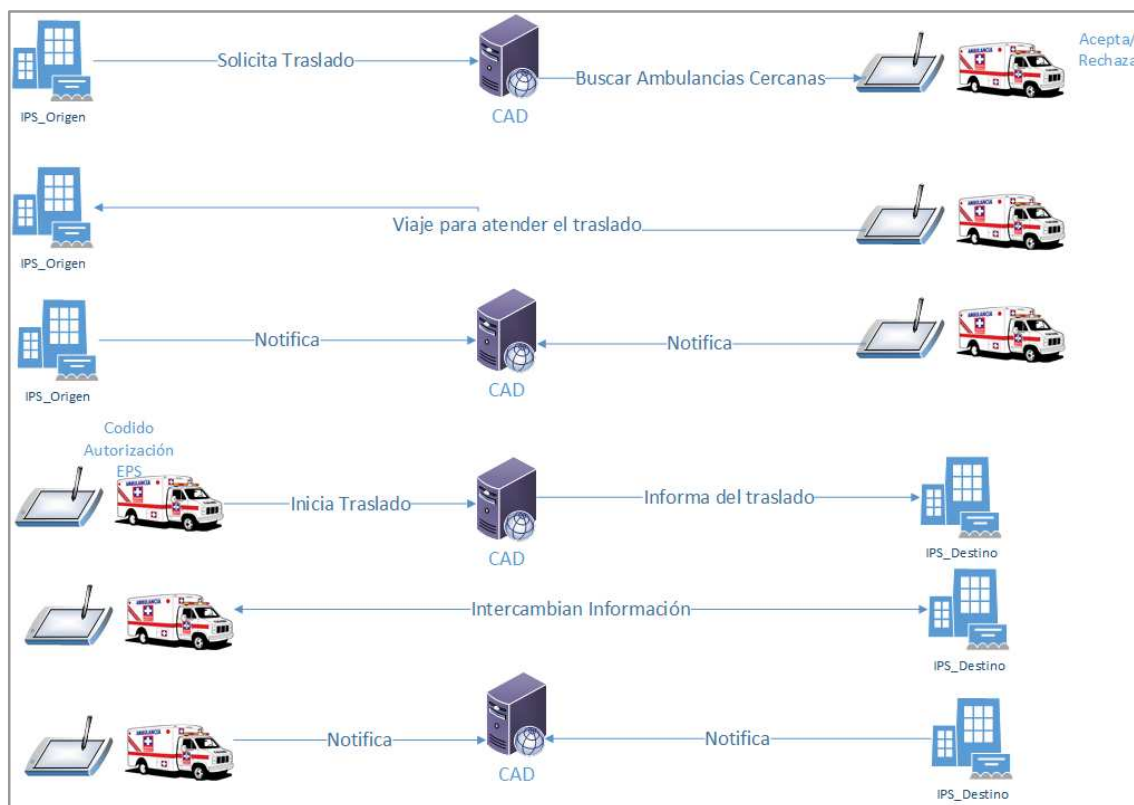


FIGURA 6. PROCESO TRASLADO INTERHOSPITALARIO

7.4. REQUERIMIENTOS FUNCIONALES

Los requerimiento funcionales, establecidos para el funcionamiento del SIEF, han sido caracterizados en componentes de Front-End y Back-End, de acuerdo con la arquitectura de la aplicación. Se encuentran separados por los tipos de usuarios que hacen uso del sistema así:

7.4.1 FRONT END

1. Usuarios
2. IPS
3. Ambulancia
4. Administradores

Requerimientos generales de usuarios:

Número	Requerimientos	Usuarios
R.1	Desplegar en la interfaz grafica el rol y nombre del usuario autenticado.	TODOS
R.2	Permitir a un usuario solicitar y cancelar un servicio de traslado.	TODOS
R.3	Evaluar la calidad del servicio prestado en una escala (0 a 5).	TODOS
R.4	Permitir el ingreso (a usuarios que deben estar registrados previamente por un administrador) y la autenticación a un usuario (para autenticación, el usuario debe ingresar: cédula y contraseña).	TODOS EXCEPTO USERADD_MOVIL
R.5	Cerrar sesión.	TODOS EXCEPTO USERADD_MOVIL

1. Usuario-UserAddMóvil

R.1 Cargar la pantalla inicial con campo de dirección habilitado para escribir, mapa de ubicación actual y opciones de tipo de emergencia.

R.2 Fijar el mecanismo de ubicación geográfica (Preguntando si la ubicación geográfica se realiza a través de su dispositivo móvil usando el GPS o el navegador.)

2. Usuario- Paramédico

R.1 Cargar la pantalla inicial con el menú de opciones: Servicios, Estado, Nuevo Paciente, Actualizar Datos y Cerrar Sesión.

R.2 Asociar una ambulancia. Para hacerlo se debe ingresar la Placa del Vehículo.

R.3 Fijar el mecanismo de ubicación geográfica (Preguntando si la ubicación geográfica se realiza a través de su dispositivo móvil usando el GPS o el navegador.)

3. Usuario-Admisiones

R.1 Cargar la pantalla inicial de la lista de traslados, la cual debe permitir: ver el traslado y ver el paciente.

R.2 Preguntar si la ubicación geográfica es través de su dispositivo móvil es usando el GPS o el navegador.

R.3 Permitir aceptar o rechazar solicitudes de atención.

4. Usuario-Portero

R.1 Cargar la pantalla inicial de la lista de traslados, la cual debe permitir: ver el traslado.

R.2 Confirmar la entrada de un paciente de la lista de traslados.

R.3 Notificar al usuario con un mensaje emergente sí un traslado se encuentra en camino.

R.4 Consultar los traslados, cada traslado tendrá la siguiente información: Mapa de la ruta, tiempo aproximado de llegada, nombre del paciente.

5. Usuario-Médico

R.1 Cargar la pantalla inicial de la lista de traslados, la cual debe permitir: ver el traslado y ver el paciente.

R.2 Consultar los traslados, cada traslado tendrá la siguiente información: Mapa de la ruta, tiempo aproximado de llegada, nombre del paciente e información médica.

R.3 Notificar al usuario con un mensaje emergente sí un traslado se encuentra en camino.

R.4 Confirmar la atención de un paciente.

R.5 Permitir enviar y recibir notas de voz de la ambulancia.

6. Administrador Empresa Ambulancia

R.1 Cargar la pantalla inicial de la lista de usuarios la cual debe permitir: ver la información del usuario, el rol y el estado.

R.2 Realizar la asignación de roles a los usuarios. Los roles permitidos son: Administradores, Paramédico.

R.3 Listar los usuarios con sus respectiva información y roles.

R.4 Permitir la creación de usuarios en el sistema con la información básica requerida para su descripción. La información básica para el registro de un usuario incluye: IDENTIFICACIÓN, PASSWORD, NOMBRES, APELLIDOS, E-MAIL, ROL.

R.5 Modificar la información básica de los usuarios del sistema y su rol en el sistema.

R.6 Eliminación de los usuarios del sistema.

R.7 Actualizar Datos de los usuarios y de la sesión.

R.8 Ingreso de información de las ambulancias, con los siguientes datos: PLACA, MODELO, MARCA, TIPO: TAB, TAM.

R.9 Generar reportes diarios y mensuales de las actividades realizadas por una ambulancia ingresando la PLACA.

R.10 Generar reportes diarios y mensuales de las actividades realizadas por un paramédico ingresando la cédula.

7. Administrador EPS

R.1 Cargar la pantalla inicial de la lista de usuarios e IPS la cual debe permitir: visualizar la información del usuario, IPS y los roles.

R.2 Asignar roles a los usuarios. El rol permitido para esta función es: Administradores.

R.3 Actualizar datos de los usuario y de la sesión.

R.4 Permitir crear EPS e IPS en el sistema con la información básica requerida para su descripción. La información básica para el registro de un usuario incluye: NIT, NOMBRE, ALIAS, E-MAIL, ROL.

R.5 Listar los usuarios e IPS con sus respectiva información y roles.

R.6 Permitir crear usuarios en el sistema con la información básica requerida para su descripción. La información básica para el registro de un usuario incluye: IDENTIFICACIÓN, PASSWORD, NOMBRES, APELLIDOS, E-MAIL, ROL.

R.7 Modificarla información de los usuarios e IPS del sistema y su rol en el sistema.

R.8 Eliminar de los usuarios e IPS del sistema.

R.9 Generar reportes diarios y mensuales de las solicitudes atendidas por IPS ingresando el NIT.

R.10 Generar reportes diarios y mensuales de las actividades realizadas por un administrador ingresando la cédula.

8. Administrador General

R.1 Permitir la creación de usuarios en el sistema con la información básica requerida para su descripción. La información básica para el registro de un usuario incluye: IDENTIFICACIÓN, PASSWORD, NOMBRES, APELLIDOS, E-MAIL, ROL.

R.2 Crear EPS e IPS en el sistema con la información básica requerida para su descripción. La información básica para el registro de un usuario incluye: NIT, NOMBRE, ALIAS, E-MAIL, ROL.

R.3 Realizar la asignación de roles a los usuarios. Los roles permitidos son: TODOS.

R.4 Modificar la información de los usuarios, IPS y EPS del sistema, además de su rol en el sistema

R.5 Listar todos los usuarios, EPS e IPS con sus respectiva información y roles.

R.6 Eliminar de los usuarios, EPS e IPS del sistema.

R.7 Actualizar datos de los usuario y de la sesión.

R.8 Generar reportes diarios y mensuales de las solicitudes atendidas por IPS ingresando el NIT.

R.9 Generar reportes diarios y mensuales de las actividades realizadas por un administrador ingresando la cédula.

R.10 Generar reportes diarios y mensuales de las actividades realizadas por una ambulancia ingresando la PLACA.

R.11 Generar reportes diarios y mensuales de las actividades realizadas por un paramédico ingresando la cédula.

7.4.2 REQUERIMIENTOS NO FUNCIONALES

Esta sección contiene los requerimientos no funcionales determinados para el funcionamiento de SIEF así como una subsección de los requerimientos no funcionales del Back-End.

Estos son los requerimientos no funcionales determinados:

1. Requerimientos de Auditoría:

R.1 Al procesar cada solicitud, se deberá dejar un reporte (log) con la información del usuario que ejecuta la solicitud, la fecha y hora de la misma.

R.2 Al procesar cada confirmación, se deberá dejar un reporte (log) con la información del usuario que ejecuta la solicitud, la fecha y hora de la misma.

2. Requerimientos de Seguridad:

R.3 Los archivos se deben almacenar usando firmas digitales, clave pública y clave privada.

R.4 Toda la información debe tener al menos un servidor de respaldo de actualización diaria.

3. Requerimientos de Desempeño

R.5 Se debe procesar un mínimo de 100 solicitudes por minuto.

4. Requerimientos de Disponibilidad

R.6 El sistema debe tener una disponibilidad mínima del 96.00%

R.7 El sistema debe tener un esquema de contingencia que evite hacer el proceso manual de la información del paciente.

5. Requerimientos de Capacidad y Escalabilidad

R.8 Se deberá soportar el crecimiento de unos 100 clientes nuevos cada mes.

R.9 Debe ser posible soportar el crecimiento del 10% en el número de solicitudes estimadas cada mes.

R.10 El sistema deberá tener una capacidad inicial de: una empresa de ambulancias, una institución médica y un usuario móvil.

6. Requerimientos de Robustez

R.11 Al procesar los reportes de empresas de ambulancia y/o instituciones médicas se realizará una consulta hasta el minuto anterior a dicha solicitud de reporte.

7. Requerimientos de Retención

R.12 La información de los usuarios deberá permanecer en el sistema por un año adicional a partir de la cancelación de la cuenta.

R.13 La información de Historia Médica de los pacientes debe ser entregada a las Instituciones médicas correspondientes al destino (quienes deberán conservar dicha información por el tiempo requerido por Ley para Historias Médicas).

R.14 Se deberá firmar un contrato de claridad de manejo y entrega de la información para todos los usuarios que participan en el sistema.

8. Requerimientos de Acceso

R.15 La interfaz de usuario (GUI, por sus siglas en inglés) debe ser de fácil uso y acceso para los diferentes usuarios, conservando los protocolos establecidos de colores y manejo de logotipos médicos.

R.16 La interfaz gráfica debe resultar fácil de utilizar e interpretar para todos usuarios.

9. Requerimientos de Velocidad y Procesamiento

R.17 La respuesta de solicitudes y mensajes de confirmación del sistema máximo debe ser de 80 segundos.

7.4.3 REQUERIMIENTOS DEL BACK-END

R.1 Identificar la ambulancia más cercana al sitio de la solicitud con: dirección origen e información de las ambulancias en servicio.

R.2 Enviar al menos a 3 ambulancias cercanas al origen, la solicitud de traslado.

R.3 Seleccionar solo la primera ambulancia que acepte el traslado

R.4 Enviar al usuario que solicita el servicio información de la ambulancia seleccionada: TIPO: TAM-TAB, PLACA, MARCA, Paramédico.

R.5 Enviar y recibir los mensajes de notificaciones de aceptación y rechazo de servicios, solicitudes, confirmación de llegadas e impresión del servicio.

R.6 Ingresar información en las bases de datos del servicio, e información de la historia clínica de los pacientes.

R.7 Si la emergencia es de TIPO: VITAL, entonces, buscar las Instituciones médicas más cercanas según la posición de la ambulancia.

R.8 Permitir seleccionar una Institución médica a dirigirse y cargar el mapa de la ruta.

R.9 Ingresar signos vitales e información del paciente a trasladar.

R.10 Si la emergencia es de TIPO: NO VITAL, entonces, permitir solicitar la atención a una Institución médica.

R.11 Agregar la información del servicio que se está llevando a cabo a la base de datos de traslados.

R.12 Actualizar la base de datos de Ambulancias Activas, cuando un usuario paramédico asocie una ambulancia a su sesión.

R.13 Integrar el uso de las librerías de Google Maps para establecer las rutas.

R.14 Actualizar la información de las ambulancias y de los traslados.

R.15 Almacenar información de las Instituciones médicas No inscritas

R.16 Almacenar Información de las Instituciones médicas inscritas.

R.17 Almacenar en diferentes fuentes de información las: Ambulancias Activas y Ambulancias Registradas.

R.18 Permitir la comunicación directa entre Ambulancia e Institución Médica una vez aceptado el servicio de traslado.

R.19 Permitir el envío de al menos una variable de información desde la Ambulancia a la Institución médica a dirigirse.

7.4.4 REQUERIMIENTOS BACK-END GENERALES: NOTIFICACIONES EMERGENTES

R1. Informar a la Institución Médica seleccionada a dirigirse, que se encuentra en camino una emergencia de Tipo Urgencia Vital.

R2. Informar a la Institución Médica seleccionada, la solicitud de atención con Documento Adjunto.

R3. Mensaje emergente del traslado en camino a: Portero, Médico y Admisiones.

7.5. DOCUMENTACIÓN DE DISEÑO

Con base en los documentos establecidos previamente, se diseñó un servicio que cumpla con los requerimientos funcionales y no funcionales, buscando dar solución al problema planteado. En esta sección, se describe el proceso de diseño de la solución dividido en 2 apartados: especificación del diseño funcional y especificación del diseño técnico.

7.6. ESPECIFICACIÓN DEL DISEÑO FUNCIONAL

En el diseño funcional, se establecieron de manera formal los procedimientos por realizar según el tipo de emergencia médica, los diseños gráficos de las aplicaciones de cada una de las partes y el flujo de comunicación entre el paciente, la ambulancia, el centro automático de despacho y el centro médico.

7.6.1 PROCESOS SEGÚN EL TIPO DE EMERGENCIA.

Para diseñar los procesos y la toma de decisiones que se dan en la propuesta de solución, se tuvo en cuenta la caracterización del problema realizada, logrando diseñar una serie de procedimientos que solucionan los problemas planteados, tales como: comunicación no concurrente, falta de registros digitales y desinformación del estado del paciente durante el traslado a una institución médica.

El diseño de los procesos está basado en los tres tipos de emergencia que pueden presentarse: urgencia vital, urgencia no vital y traslado interhospitalario y la forma cómo deben interactuar los agentes del servicio (Paciente, IPS, CAD y Ambulancia), posibilitando una comunicación eficiente. Con este diseño, es posible formalizar el esquema funcional de la propuesta de solución, a través de un diagrama de procesos claro. Una ventaja que presenta la propuesta de solución es utilizar una plataforma en la nube, como Google Cloud, para desarrollar e implementar la funcionalidad del Centro Automático de Despacho (CAD); ya que se puede tener acceso a los servicios del Back-End desde cualquier lugar, además, al contar con el respaldo de la infraestructura de Google, la información crucial para el funcionamiento del servicio, siempre estará disponible y protegida de cualquier eventualidad.

Lo procesos a la hora de tener una Urgencia Vital son:

1. El paciente inicia el proceso solicitando al CAD una ambulancia para atender una emergencia.
2. El CAD se encarga de ubicar las 3 ambulancias más cercanas al lugar del evento y les envía una notificación.
3. Luego que la ambulancia llega, el paramédico o el médico responsable de los traslados de la ambulancia informa al CAD si se trata o no de una urgencia vital.
4. El CAD buscará las 3 instituciones más cercanas para iniciar el traslado; una vez elegida la institución destino, el CAD informará al personal de la entidad de salud sobre el traslado y las condiciones del paciente.
5. Una vez el paciente llega a la institución, se envía una notificación y la información del traslado se almacena de manera permanente en la base de datos.

A continuación, se presentan los procesos para la atención de una urgencia no vital:

1. En caso de ser una urgencia no vital, nuevamente la interacción es iniciada por el paciente, quien solicita una ambulancia al CAD.
2. En el momento que la ambulancia llega al lugar y se determine que la urgencia es no vital, el paramédico puede documentar al paciente y solicitar el cuidado del paciente en una institución médica que tenga convenio con la EPS del paciente.
3. Si la respuesta es positiva, el traslado se inicia, en caso contrario, se buscará otro centro médico.
4. Una vez el paciente llega a la institución, se envía una notificación y la información del traslado se almacena de manera permanente en la base de datos.

Por último, se presenta el diagrama de procesos del traslado interhospitalario:

1. En este último caso, la solicitud es iniciada por una IPS de origen, la cual necesita una ambulancia para trasladar al paciente a otra IPS.
2. Una vez la ambulancia llega al sitio, el paciente ya se encuentra documentado y el traslado es iniciado, según el nivel de urgencia.
3. Con respecto a este tipo de solicitud, al ser en su mayoría programados, ya se tiene un código de autorización de la EPS se evitan problemas en la admisión del paciente en el IPS destino.

Para ver de manera más detallada los diagramas de procesos desarrollados, puede dirigirse al anexo "*TiposdeEmergencias.vsd*", donde encontrará los procesos descritos en esta sección expuestos en una serie de diagramas que permiten comprender mejor el funcionamiento de la herramienta. Este entregable es la parte crucial del diseño funcional, ya que define los procesos necesarios para atender cada uno de los tipos de emergencia que pueden darse en el momento de solicitar atención médica, utilizando la solución propuesta.

7.6.2 MODELOS GRÁFICOS INTERFAZ DE USUARIO (MOCK-UPS)

Este proyecto debe brindar una solución para usuarios finales que resulte fácil de utilizar e interpretar, además de ser intuitiva y con una interfaz gráfica amigable, la cual, a través de iconos y poco texto pueda brindarle al usuario una grata experiencia a la hora de utilizar el servicio propuesto. Debido a esto, se realizaron los diseños de las interfaces gráficas de usuario de cada una de las aplicaciones que utilizan los agentes involucrados.

Estos diseños tienen como objetivo ser las maquetas (mock-ups) del servicio, para poder hacer una evaluación previa con las personas involucradas y determinar si los diseños cumplen con el objetivo de ser intuitivos y simples. Con estos Mock-ups, se buscó plasmar, mediante la interfaz gráfica, un primer vistazo de la solución y como a través de la utilización del servicio móvil se resuelven los requerimientos funcionales y no funcionales. En total, se elaboran más de 110 vistas diferentes, combinadas entre mock-ups de aplicaciones móviles de usuario-paciente, ambulancia, vigilante y médico; además de tener una aplicación web para el personal de admisiones y administradores del servicio.

Algo importante que debe mencionarse, es que se consideró realizar una aplicación dedicada al vigilante de la puerta de urgencias, ya que al realizar la caracterización del problema se logra identificar que este agente es un cuello de botella dentro de los procedimientos actuales, por lo que es necesario que dicha persona conozca, qué traslados se dirigen hacia la institución médica. A continuación se presentan algunos ejemplos de las vistas desarrolladas para cada mock-up, por obvias razones en este documento no se presentan todas las vistas hechas; si se desea ver en su totalidad las vistas realizadas puede dirigirse al anexo “*Diseño Interfaces Gráficas.zip*”: (Ver figuras del 7 al 12).



FIGURA 7. MOCK-UP APP USUARIO-PACIENTE



FIGURA 8. MOCK-UP APP AMBULANCIA

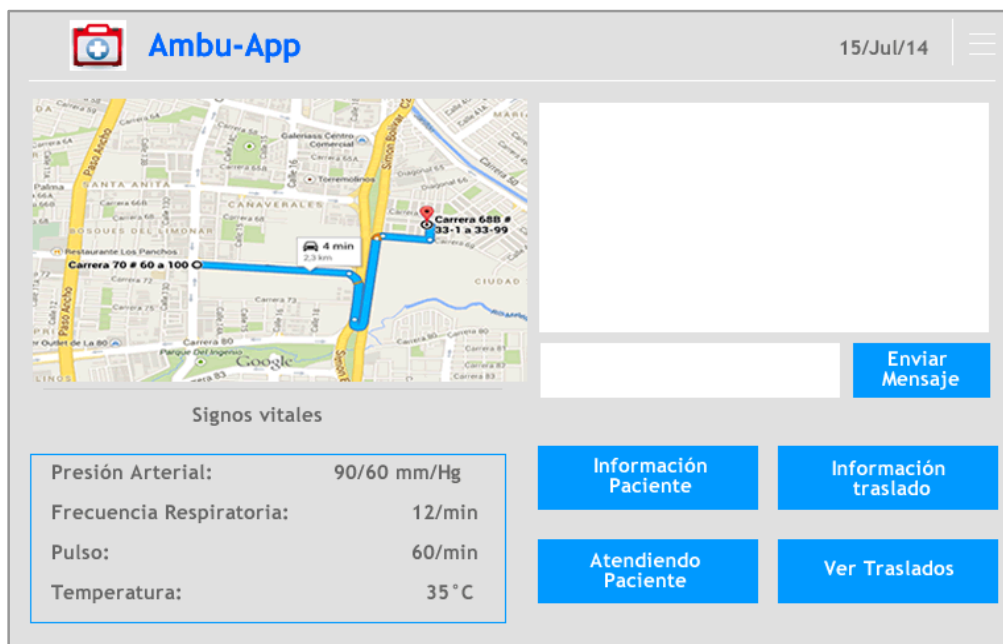


FIGURA 9. MOCK-UP APP MÉDICO


Ambu-App
15/Jul/14

Lista de Traslados

Confirmar Ingreso	Paciente	Estado	Resumen Urgencia	Tipo Urgencia	Nivel Urgencia	Responsable	Acompañante	Traslado
<input type="checkbox"/>	Juan Felipe	En Camino	Traslado para realizar examen	SOAT	1	SOAT	_____	Ver
<input checked="" type="checkbox"/>	Angela	Paso a Admisiones	Dolor Cabeza	Domestica	3	Comfenalco	_____	Ver
<input type="checkbox"/>	Wilson	En Espera	Traslado para realizar examen	Traslado Inter-Hospital	4	Cruz Blanca	Ana	Ver

Aceptar

FIGURA 10. MOCK-UP APP VIGILANTE


Home
Traslados
Solicitar Ambulancia
Cambiar Contraseña
More

Traslados

Paciente	Estado	Resumen Urgencia	Tipo Urgencia	Nivel Urgencia	Responsable	Traslado	Paciente
Juan Felipe Gómez	En Camino	Realizar Examen Médico	Traslado Inter-Hospital	4	Sanitas	Ver	Ver
Luz Angela Sarria	Admitido	Dolor de Cabeza	Doméstica	4	Comfenalco	Ver	Ver

FIGURA 11. MOCK-UP APLICACIÓN WEB ADMISIONES


Iniciar Sesión
Admin Maestro
Admin Institución Médica
More

Administradores

Instituciones Médicas

Empresas Ambulancias

FIGURA 12. MOCK-UP APLICACIÓN WEB ADMINISTRADORES DEL SISTEMA

En esta fase del desarrollo del proyecto, se definió que el nombre de la solución propuesta sería “Ambu-app”, pero al realizar un análisis del producto final, se determinó que el núcleo del servicio no eran las ambulancias, sino la gestión en la atención de emergencias médicas, la cual, al ser integrada con el intercambio de información en el momento de traslado de un paciente, se lograba tener un sistema funcional y eficiente; por lo que se decidió cambiar el nombre a Sistema Integrado de Emergencias Funcional (SIEF).

7.6.3 DIAGRAMA DE NAVEGACIÓN

Una vez culminada la etapa del diseño de la interfaz gráfica, y de realizar las pruebas de aceptación por parte de los usuarios, fue necesario desarrollar un diagrama de navegación para cada una de las aplicaciones propuestas, con el fin de formalizar la interacción entre las vistas diseñadas. Con el diagrama de navegación, es posible seguir, de una manera más simple, la secuencia de ventanas a la hora de presionar un botón o recibir una notificación, de esta manera es posible verificar la forma como con la interfaz gráfica satisface con los requerimientos funcionales impuestos. Un diagrama de navegación se considera un entregable importante porque formaliza la elaboración de una interfaz gráfica funcional y de sencilla interacción para el usuario, además, se puede evaluar la cantidad de acciones requeridas para cumplir una tarea como cambiar la contraseña, ver los traslados, actualizar signos vitales, entre otros. A continuación, se muestra el diagrama de navegación para la aplicación del usuario, figura 13.

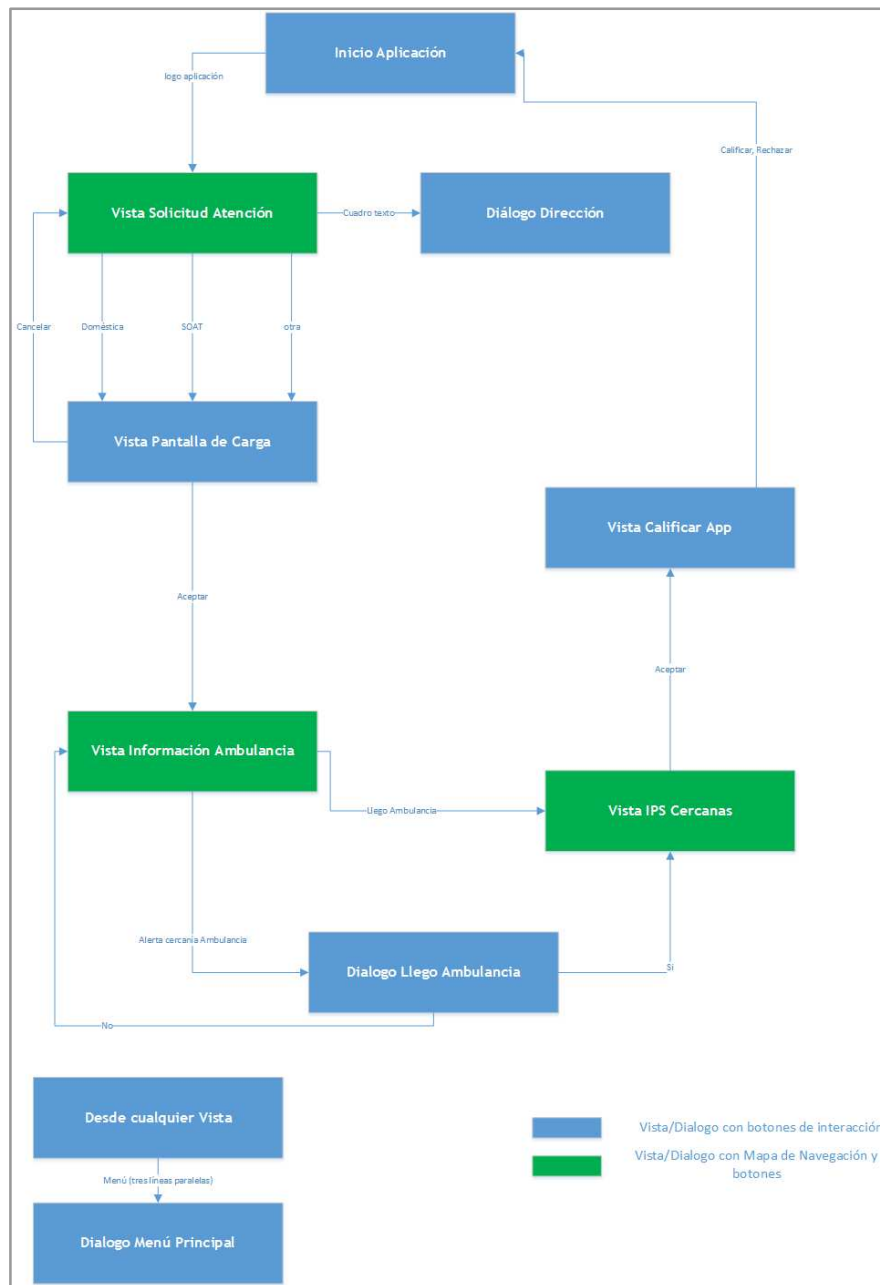


FIGURA 13. DIAGRAMA DE NAVEGACIÓN

Desde este diagrama de navegación es posible apreciar la forma como el usuario de la aplicación móvil, solicita la atención de una emergencia, ver la información de la ambulancia, notificar la llegada de la ambulancia e identificar las instituciones médicas cercanas.

En el diagrama de los paramédicos, se aprecia la forma como el usuario interactúa con su información de usuario, visualiza el historial de traslados de la ambulancia, determina el tipo de urgencia, documenta al paciente, transmite información a la institución médica destino entre otras actividades.

En cuanto al médico, en el diagrama de navegación se establece que el médico puede cambiar la contraseña, ver la información del traslado, la información del paciente, la ubicación de la ambulancia, terminar la atención del paciente, entre otras actividades.

Por último se tiene al vigilante, quien solo puede interactuar con la aplicación viendo los traslados programados, cambiar su contraseña e indicar que el paciente se encuentra en la institución.

Con estos diagramas es más claro entender la forma cómo interactúan los agentes con cada una de las aplicaciones diseñadas para ellos. Para ver todos los diagramas realizados con mayor detalle, observar el anexo *“DiagramadeNavegación.vsdX”*

7.7. ESPECIFICACIÓN DEL DISEÑO TÉCNICO

Una vez se tiene un diseño funcional claro, en el cual se ha definido el funcionamiento genérico del servicio y la forma como se da la interacción entre cada persona involucrada en la atención de una emergencia, es posible determinar la forma como la propuesta de solución resolverá el problema planteado. En esta sección del documento, se describe el proceso de diseño técnico (la etapa de ingeniería), el cual será la base para poder hacer realidad todo lo planteado en los diseños funcionales y en la fase del análisis.

7.7.1 MODELO DE CONCEPTOS

En esta sección, se menciona el desarrollo de un modelo de conceptos, que constituye la base para el modelo relacional y el diagrama de clases. El modelo de conceptos permite realizar una abstracción del problema, identificando los elementos más relevantes que tienen influencia en la elaboración de la propuesta de solución. Para poder realizar dicha abstracción es necesario tener muy claro la problemática actual y los diseños funcionales, descritos previamente, así pues, los siguientes son los conceptos claves de la solución:

1. **Instituciones:** Los diferentes tipos de institución que tienen participación en el sistema, tales como instituciones médicas, empresas de ambulancias y entidad administradora del sistema -administradores de SIEF- Se decidió unir estos tres tipos de institución en un solo concepto, debido a la similitud entre sus atributos.
2. **Empleados:** Hace referencia a las personas que trabajan en las instituciones definidas. Éstas pueden tener un rol diferente dentro de las instituciones, y dependiendo del rol, se define qué aplicación puede utilizar.
3. **Rol:** Corresponde con los diferentes roles que pueden tener los empleados de una institución, dependiendo del Rol, se asigna un perfil al empleado, y éste sólo podrá ejecutar la aplicación a la que tenga permiso, así, un vigilante no podrá ejecutar la aplicación móvil diseñada para los médicos.
4. **Ambulancias:** Responsable por realizar los traslados del paciente y por responder a las solicitudes que estos generan. Pueden ser tipo TAB y TAM, pero siempre tienen una persona que interactuará con la aplicación móvil.

5. **Ubicaciones:** Dentro del servicio, se ha considerado guardar un registro de las actividades, de esta manera, se busca tener un historial de ubicaciones de las ambulancias, así, la búsqueda será más eficiente y siempre se sabrá la ubicación de las ambulancias inscritas en el sistema.
6. **Paciente:** Es la persona a quien se le presta el servicio de atención médica y se le hace seguimiento desde el momento que solicita el cuidado hasta el momento que el médico determina que dicho paciente ha sido atendido satisfactoriamente.
7. **Enfermedad:** Corresponde con las afecciones y padecimientos que tiene el paciente. Un paciente puede tener varias enfermedades.
8. **Procedimientos:** Son los procedimientos que se han realizado al paciente para estabilizarlo y lograr manejo de la enfermedad.
9. **EPS:** Es la Entidad Promotora de Salud, la cual juega un papel crucial en la admisión de un paciente en una institución médica.
10. **Traslado:** Es el núcleo de la propuesta de solución, ya que este concepto es donde se puede entablar comunicación entre la ambulancia y la institución médica, guardando un registro de las condiciones del paciente durante el traslado.

Una vez establecidos los conceptos de la solución, es necesario representar, a través de un diagrama como se relacionan entre sí. A continuación se presenta este diagrama explicativo. Figura 14.

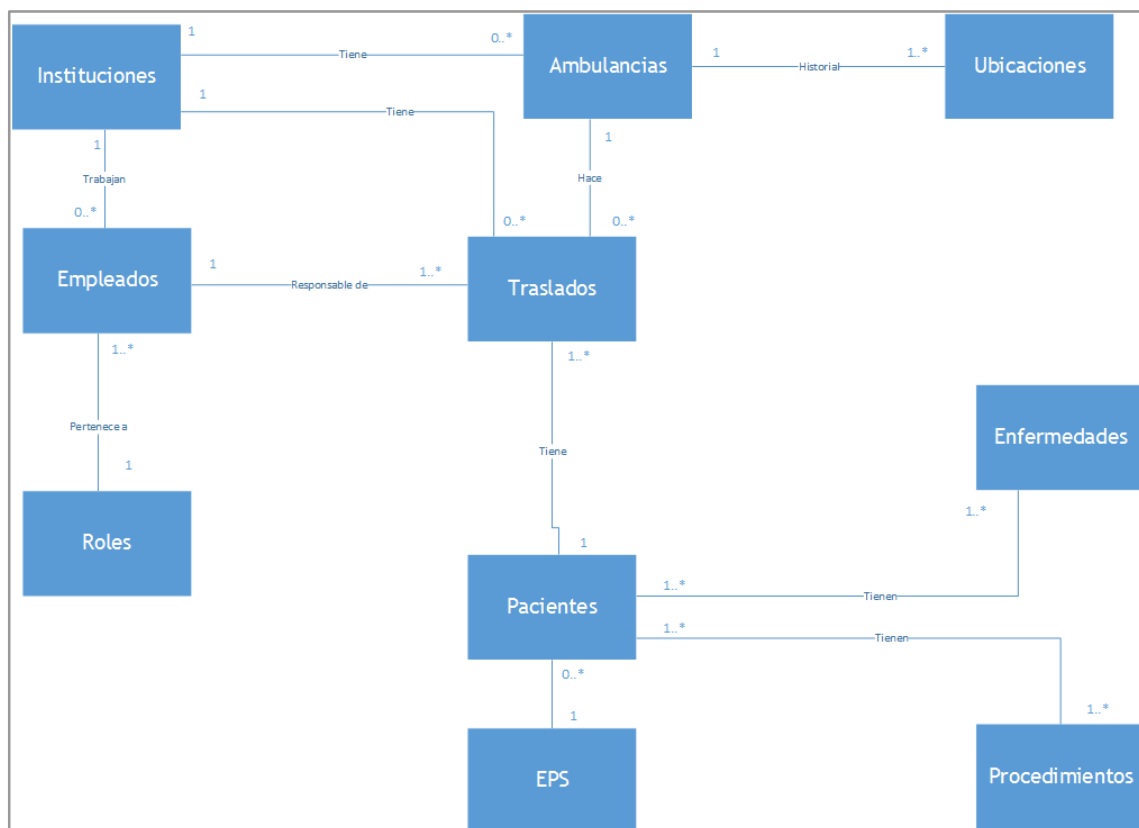


FIGURA 14. DIAGRA DE RELACIÓN ENTRE CONCEPTOS.

En este modelo de conceptos, se pueden apreciar las relaciones existentes entre cada uno de los elementos que componen la propuesta de solución. Este modelo formaliza la abstracción del esquema funcional y será la base para la implementación de las aplicaciones de usuario final (FrontEnd) y la implementación del servidor (BackEnd), en él se definen las relaciones, dependencias, estructura de datos, modelo relacional de la base de datos, diagrama de clases, intercambio de información, y en general, el funcionamiento lógico del sistema. Para ver el modelo de conceptos de manera detallada, puede ver el anexo “*Modelo Relacional.vsd*”.

Este entregable es uno de los más importantes para el diseño técnico de la solución, ya que con él es posible formar una estructura sólida para las siguientes fases del diseño, además, es parte crucial para tener una implementación de software alineada a resolver el problema planteado.

7.7.2 DISEÑO BASE DE DATOS

El siguiente paso después de tener un modelo de conceptos, es realizar el modelo relacional de la base de datos, es decir, diseñar la base de datos. Se decidió utilizar un modelo relacional debido a que organiza los datos en grupos de tablas, las cuales se relacionan por el tipo de datos que contienen, conservando las relaciones definidas en el modelo de conceptos. Este tipo de modelo permite tener de una manera clara y organizada la información necesaria para que el sistema funcione correctamente. Las características principales de una base de datos relacional son las siguientes:

- No pueden existir 2 tablas con el mismo nombre ni registro.
- Cada tabla es un conjunto de registros.
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de claves primarias y foráneas.
- Las llaves primarias son la clave principal de un registro dentro de una tabla.
- La llave foránea se colocan en la tabla hija y contiene el mismo valor que la clave primaria del registro padre.

De esta manera, se definieron las siguientes tablas para la base de datos, cada una con sus atributos:

	Nombre Tabla:	Institución		
	Descripción:	Las instituciones que tienen participación en el servicio		
Llave	Nemónico Campo	Descripción	Tipo	Longitud
Primary Key	ID_NIT	ID único cada una de las instituciones que tienen participación en el servicio	CHAR	20
	Nombre	El nombre completo con el que se identifica a la institución Médica	CHAR	20

	Razón Social	Nombre completo del profesional independiente o nombre o razón social de la IPS, como aparece en el registro de habilitación	CHAR	60
	e-mail	Correo electrónico usado por la IPS para el contacto	CHAR	30
	Dirección	Dirección donde se encuentra ubicada la Institución	CHAR	30
	Teléfono	Teléfono de Contacto de la Institución	CHAR	12
	Tipo	Campo que se utiliza para definir el tipo de institución que participa en el servicio	CHAR	2

TABLA 1. BASE DE DATOS INSTITUCIÓN

	Nombre Tabla:	Empleado		
	Descripción	Personas en el servicio		
Llave	Nemónico Campo	Descripción	Tipo	Longitud
Primary Key	Cédula	La cédula que identifica a la persona de manera única	CHAR	12
Foreign Key	ID_NIT	ID único cada una de las instituciones que tienen participación en el servicio	CHAR	20
	NOMBRE	El nombre de la persona	CHAR	30
	APELLIDO	El apellido de la persona	CHAR	30
	CONTRASEÑA	La contraseña de la persona para iniciar sesión en el servicio	CHAR	30
	SEXO	El sexo de la persona	CHAR	1
	E_MAIL	Correo electrónico usado por la persona para el contacto	CHAR	30
Foreign Key	ID_Rol	El ID del Rol que identifica el Rol del empleado	INT	2

TABLA 2. BASE DE DATOS EMPLEADO

	Nombre Tabla:	Roles		
	Descripción	Los Roles que se pueden definir dentro del servicio		
Llave	Nemónico Campo	Descripción	Tipo	Longitud
Primary Key	ID_Rol	El ID único para el Rol	Int	2
	Desp_Rol	Descripción que define al rol. Puede ser los siguientes: Paramedico, Medico, Admin Maestro, Admisiones, Vigilante	CHAR	20

TABLA 3. BASE DE DATO DE LOS ROLES

	Nombre Tabla:	Ambulancias		
	Descripción:	Las ambulancias que están en el servicio		
Llave	Nemónico Campo	Descripción	Tipo	Longitud
Primary Key	PLACA	La placa de la ambulancia	CHAR	6
Foreign Key	ID_NIT	ID único cada una de las instituciones que tienen participación en el servicio	CHAR	20
	TIPO	El tipo de ambulancia	CHAR	3
	MODELO	Modelo de la ambulancia	CHAR	4
	MARCA	La marca de la ambulancia	CHAR	18
	DISPONIBLE	Indica si la ambulancia se encuentra disponible o no	BOOLEAN	1
	UBICACION	La última ubicación conocida de la ambulancia.	CHAR	30

TABLA 4. BASE DE DATOS AMBULANCIAS

	Nombre Tabla:	Ubicaciones		
	Descripción:	Las ubicaciones registradas en el servicio.		
Llave	Nemónico Campo	Descripción	Tipo	Longitud
Primary Key	ID_UBICACION	El identificador único para la ubicación.	int	6
Foreign Key	PLACA	La placa de la ambulancia asociada a la ubicación	CHAR	6
	LONGITUD	El parámetro longitud de la coordenada.	CHAR	14
	LATITUD	El parámetro latitud de la coordenada.	CHAR	4

TABLA 5. BASE DE DATOS DE UBICACIONES

	Nombre Tabla:	Paciente		
	Descripción:	La tabla que tiene la información del paciente		
Llave	Nemónico Campo	Descripción	Tipo	Longitud
Primary Key	DOCUMENTO_IDENTIDAD	El número del documento que identifica al paciente de manera única	CHAR	12
	NOMBRE	El nombre de la persona	CHAR	30
	APELLIDO	El apellido de la persona	CHAR	30
	SEXO	El sexo de la persona	CHAR	1
	TipoID	Tipo del documento de identidad del paciente. Puede ser Cédula, Tarjeta de identidad.	CHAR	20
	Edad	La edad del paciente	CHAR	2

Foreign Key	Cod_EPS	El código único que identifica a la EPS	CHAR	6
	Tipo_Usuario	El tipo de usuario de la EPS. Puede ser 1,2,3	CHAR	2
	Rango	El rango del usuario de la EPS	CHAR	1
	Cod_autEPS	El código de autorización de la EPS para el traslado interhospitalario	CHAR	12
Foreign Key	Cod_Enfermedad	El código único de la enfermedad, por la cual está siendo atendido el paciente	CHAR	4
Foreign Key	Cod_Procedimiento	Código único del procedimiento que se le realizó al paciente	CHAR	6
	PRESIÓN ARTERIAL	La presión de arterial del paciente. Unidades mm/Hg	INT	
	FRECUENCIA RESPIRATORIA	La frecuencia respiratoria del paciente. Unidades /min	INT	
	PULSO	El pulso del paciente. Unidades /min	INT	
	TEMPERATURA	La temperatura del paciente. Unidades °C	INT	

TABLA 6. BASE DE DATOS PACIENTE

	Nombre Tabla:	Enfermedades		
	Descripción:	Clasificación Internacional de Enfermedades y Problemas relacionados con la salud.		
Llave	Nemónico Campo	Descripción	Tipo	Longitud
Primary Key	COD_ENF	Código con el que se identifica la enfermedad o el problema de salud.	CHAR	4
	DESP_ENF	Descripción de la enfermedad o del problema de salud	CHAR	256

TABLA 7. BASE DE DATOS DE ENFERMEDADES

	Nombre Tabla:	Paciente/Enfermedad		
	Descripción:	La tabla puente entre el paciente y enfermedades		
Llave	Nemónico Campo	Descripción	Tipo	Longitud
Primary Key	COD_ENF	Código con el que se identifica la enfermedad o el problema de salud.	CHAR	4
Primary Key	DOCUMENTO_IDENTIDAD	El número del documento que identifica al paciente de manera única	CHAR	12

TABLA 8. BASE DE DATOS PACIENTE/ENFERMEDAD

	Nombre Tabla:	Procedimientos		
	Descripción:	Códigos Únicos de Procedimientos en Salud		
Llave	Nemónico Campo	Descripción	Tipo	Longitud
Primary Key	COD_PROCE	Código con el que se identifica el procedimiento.	CHAR	6
	DESP_PROCE	Descripción del procedimiento.	CHAR	256

TABLA 9. BASE DE DATOS DE PROCEDIMIENTOS

	Nombre Tabla:	Paciente/Procedimiento		
	Descripción:	La tabla puente entre el paciente y enfermedades		
Llave	Nemónico Campo	Descripción	Tipo	Longitud
Primary Key	COD_PROCE	Código con el que se identifica el procedimiento.	CHAR	6
Primary Key	DOCUMENTO_IDENTIDAD	El número del documento que identifica al paciente de manera única	CHAR	12

TABLA 10. BASE DE DATOS PACIENTE/PROCEDIMIENTO

	Nombre Tabla:	EPS		
	Descripción:	Empresas Promotoras de Salud		
Llave	Nemónico_Campo	Descripción	Tipo	Longitud
Primary Key	COD_EPS	Código con el que se identifica la EPS.	CHAR	6
	NOM_EPS	Nombre de la EPS	CHAR	60
	RazonSocial	Nombre completo del profesional independiente o nombre o razón social de la IPS, como aparece en el registro de habilitación	CHAR	60

TABLA 11. BASE DE DATOS EPS

	Nombre Tabla:	Traslado		
	Descripción:	Los traslados que se realizan en el servicio		
Llave	Nemónico_Campo	Descripción	Tipo	Longitud
Primary Key	ID_traslado	La identificación única del traslado	CHAR	6
Foreign Key	PLACA	La placa de la ambulancia	CHAR	6
Foreign Key	ID_NIT	ID único cada una de las instituciones que tienen participación en el servicio	CHAR	20

Foreign Key	DOCUMENTO_IDENTIDAD	El número del documento que identifica al paciente de manera única	CHAR	12
	Tipo	El tipo de traslado: SOAT, Domestico, Traslado Interhospitalario	CHAR	26
	NIVEL URGENCIA	El nivel de urgencia del traslado	CHAR	1
	ESTADO	El estado en el que se encuentra el Traslado. Puede ser: En camino, En espera, Paso a Admisiones, Atendido	CHAR	8
	Dir_Origen	La dirección de origen del traslado	CHAR	30
	Dir_Destino	La dirección de destino del traslado	CHAR	30
	Fecha_Init	La fecha y hora de inicio del traslado	DateTime	
	Fecha_Fin	La fecha y hora final del traslado	DateTime	
	COD_AUTORIZACION EPS	El código de autorización de la EPS que permite identificar que el traslado ha sido autorizado.	CHAR	12
Foreign Key	Responsable	La cédula del responsable del traslado en la ambulancia.	CHAR	12

TABLA 12. BASE DE DATOS TRASLADO

Como se tiene una relación de muchos a muchos entre el paciente-enfermedades y paciente-procedimientos, es necesario crear una tabla en la base de datos para conservar esta relación, ya que para un gestor de base de datos como MySQL u Oracle no es posible establecer un vínculo de muchos a muchos. Las tablas destinadas para conservar esta relación son: Paciente/Enfermedad y Paciente/Procedimiento, las cuales harán la función de puente entre el paciente y sus enfermedades y procedimientos, sin afectar que una enfermedad y/o procedimiento pueda ser aplicada a diferentes pacientes.

De esta manera, al definir las tablas y sus atributos, es posible formalizar el modelo relacional de la base de datos y diseñar la interacción entre cada uno de los elementos del sistema. Debido a la complejidad del modelo relacional de la base de datos, no es posible mostrarlo en un diagrama simplificado, por lo que se incluye en el anexo “*Modelo Relacional.vsd*”.

Un último elemento importante para terminar el diseño de la base de datos, fue la elección del gestor de la base de datos. En el mercado, existen una gran cantidad de programas que manejan los datos y ayudan en los procesos de adición, búsqueda, remoción y modificación de la información contenida en la base de datos; pero se destacan Oracle, MySQL y SQLServer. Para poder elegir el que mejor compatibilidad tuviera con la propuesta de solución, se realizó un documento comparativo, entre los 3 gestores principales, del cual se obtuvo la tabla 13:

Característica	Oracle	MySQL	SQLServer
Interfaz Gráfica	GUI,SQL	SQL	GUI,SQL
Lenguaje Soportado	C, C#, C++, Java, Ruby, Objective C	C, C#,C++, Java, Ruby, Objective C	Java, Ruby, Python, VB, .NET, PHP
Sistema Operativo	Windows, Linux, Solaris, HP-UX, OS X, z/OS, AIX	Windows, Linux, OS X, FreeBSD, Solaris	Windows
Licencia	Propietario	Código Libre	Propietario

TABLA 13. TABLA COMPARATIVA

Después de contrastar cada una de las opciones estipuladas, se determinó que el gestor a utilizar sería MySQL debido a estar basado que puede soportar el lenguaje de programación Java, el cual es el lenguaje elegido para implementar la solución, tanto en el FrontEnd como en el BackEnd; además, es un gestor tipo Código Abierto (Open Source) de licencia gratuita por lo que no se tienen gastos para su utilización, lo que representa una gran ventaja para desarrollar un proyecto de grado. Otro elemento relevante en la elección de MySQL, es la integración con el DataStore de Google Cloud, así, será posible implementar localmente la base de datos y posteriormente, subirla a la plataforma en la nube sin necesidad de cambiar nada en su estructura.

Para ver con mayor detalle la información presentada en esta sección puede dirigirse a los documentos anexos: *"TablasMaestra - Base de datos.xlsx"* y *"Caracterización Bases de Datos.docx"*.

Al terminar esta fase, es posible tener una estructura sólida de base de datos que permita desarrollar una solución en la cual almacenar la información generada por los conceptos definidos previamente, y así cumplir los requerimientos funcionales planteados y lograr tener un registro de las interacciones entre los agentes del servicio, permitiendo facilitar labores de auditoría.

7.7.3 MODELO DE MENSAJES

Un elemento importante dentro del desarrollo del sistema es establecer un protocolo de comunicación que permita la interacción ordenada de las diferentes partes involucradas en el servicio móvil. Es importante consolidar un protocolo de comunicación claro que permita tener una comunicación coordinada y eficiente entre cada uno de los agentes implicados en la solución propuesta, y así, poder garantizar que los diseños funcionales se cumplan.

De esta manera, basándose en la fase del diseño funcional del SIEF, se diseñó un modelo de mensajes simple que permitiera una interacción eficiente entre el usuario-paciente,

paramédico, CAD, médico y vigilante; garantizando una comunicación concurrente y eficiente a través de un canal seguro de comunicación.

A continuación, se presenta el modelo de mensajes que representa el protocolo de comunicación definido. Figura 15.

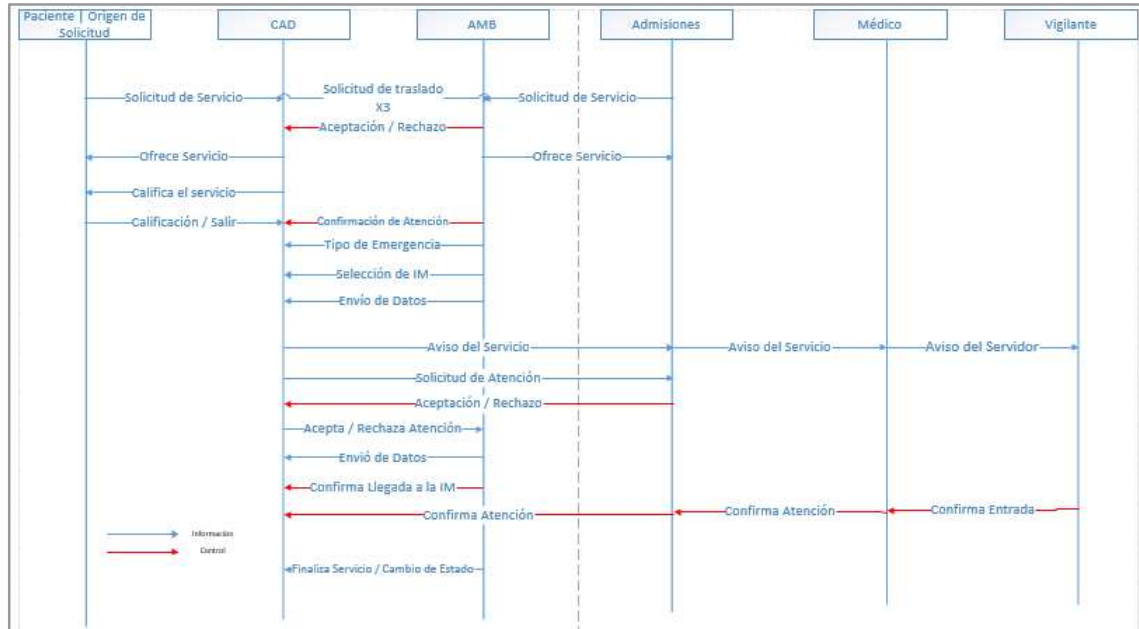


FIGURA 15. PROTOCOLO DE COMUNICACIÓN

En el diagrama, se puede apreciar que se tienen 2 tipos diferentes de mensajes: mensajes de información (Color Azul) y mensajes de control (Color Rojo). El primer tipo de mensajes hace referencia a los mensajes que transmiten alguna información relevante para el correcto funcionamiento del servicio, dicha información corresponde con: las coordenadas de los diversos agentes involucrados, los signos vitales del paciente, el resultado de una búsqueda, entre otras; gracias a esta información la comunicación entre las partes involucradas será funcional y se logra cumplir con los requerimientos funcionales planteados.

Para este tipo de mensajes fue necesario especificar qué información estará contenida en cada uno de los mensajes, ya que en un solo mensaje se pueden transmitir varios tipos de datos como cadenas de caracteres, números, un objeto java e incluso un arreglo de datos -usado para la transmisión de enfermedades del paciente, procedimientos hechos al paciente y signos vitales. Por ende, es muy importante definir un protocolo que permita que dichos datos puedan ser enviados al servidor, éste los procese y le envíe una respuesta al solicitante. Para cumplir con lo propuesto, se utiliza un protocolo denominado Simple Object Access Protocol (SOAP), el cual es un protocolo estándar que define la forma en que dos objetos en diferentes procesos, dispositivos y aplicaciones pueden comunicarse por medio de intercambio de datos XML, a través del uso de argumentos. Dicho protocolo tiene una fuerte relación con servicios web, ya que permite tener una comunicación entre diferentes plataformas sin importar su sistema operativo o requerimientos técnicos, así, se decide utilizar un servicio web con SOAP para el intercambio de mensajes entre los agentes con participación en el sistema. El protocolo de comunicación que utilizaría SIEF se formalizó en una serie de diagramas que muestran con

mayor el tipo y orden de información que se transmitiría entre las partes. A continuación, se presenta el diagrama de comunicación entre el usuario final y el servicio web ofrecido por el sistema. Figura 16.

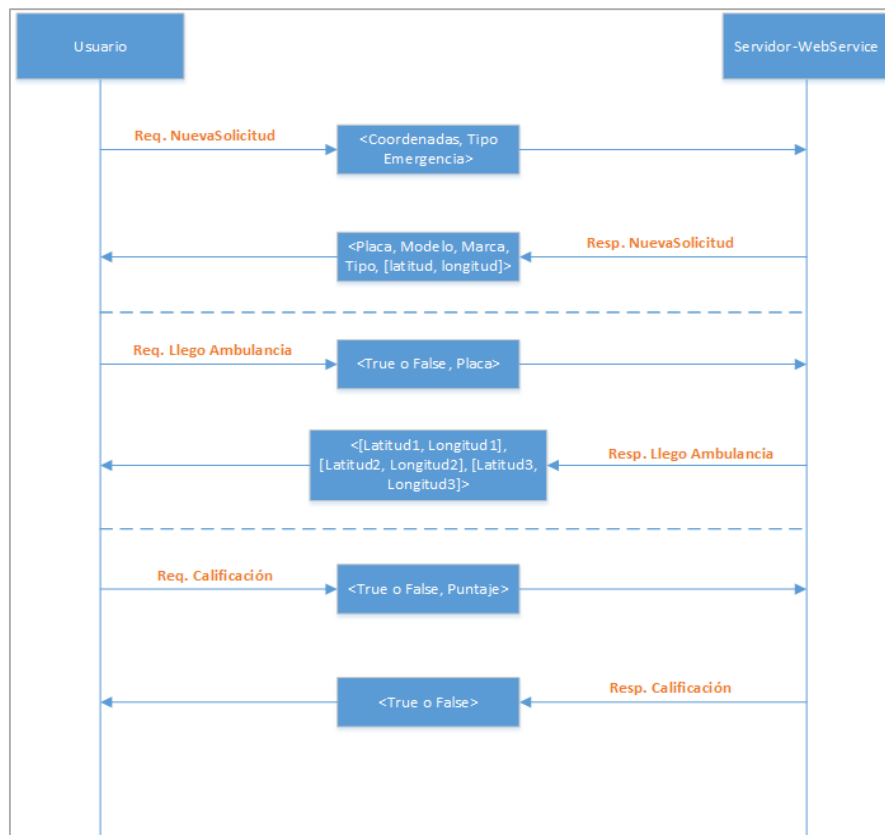


FIGURA 16. DIAGRAMA DE COMUNICACIÓN USUARIO FINAL Y WEBSERVICES

Para ver todos los diagramas planteados para el intercambio de mensajes y etiquetas XML, puede dirigirse al documento de anexo “*Diagrama Comunicación XML.vsd*”.

El segundo tipo de mensajes está destinado al control del protocolo de comunicación y a la gestión de las notificaciones entre las diferentes aplicaciones comprometidas en el sistema. En este tipo de mensaje, no se intercambia información relevante de la aplicación, ya que su uso es netamente destinado al control del servicio.

Al terminar esta fase del desarrollo del proyecto, es posible tener un protocolo de comunicación que garantiza una interacción eficiente, concurrente y funcional entre cada una de las partes involucradas, además, de lograr plantear el esquema funcional de una manera formal utilizando los diagramas presentados. Para ver el diagrama de mensajes con mayor detalle puede verse el anexo “*DiagramaMensajes.vsd*”.

7.7.4 DIAGRAMA DE CLASES

Una parte importante para el diseño de un servicio móvil como el propuesto, es el diseño de las clases que representan los agentes involucrados del servicio en una serie de aplicaciones de software, las cuales, a través de diferentes métodos, podrán ejecutar las tareas requeridas para cumplir con los requerimientos planteados. En esta fase de desarrollo, se establece la estructura del sistema propuesto para implementar una solución de software que logre hacer realidad lo planteado en los diseños anteriores. Se realizaron diferentes diagramas de clases para cada una de las aplicaciones móviles propuestas, aplicaciones web y servidor CAD; de esta manera se tendrá una guía para iniciar la implementación y, así, tener un mejor orden y desempeño en el desarrollo de la solución.

La arquitectura elegida para el desarrollo del proyecto, fue una arquitectura cliente-servidor, para poder mantener una comunicación eficiente y ordenada entre las diferentes aplicaciones que se van a implementar. En la parte del cliente, se tienen 4 aplicaciones principales, las cuales contienen sus propias clases para garantizar el buen funcionamiento de la herramienta; en la parte del servidor, solo se tiene el proyecto del CAD, el cual será implementado en la plataforma de google cloud. Para mejor entendimiento ver figura 17.

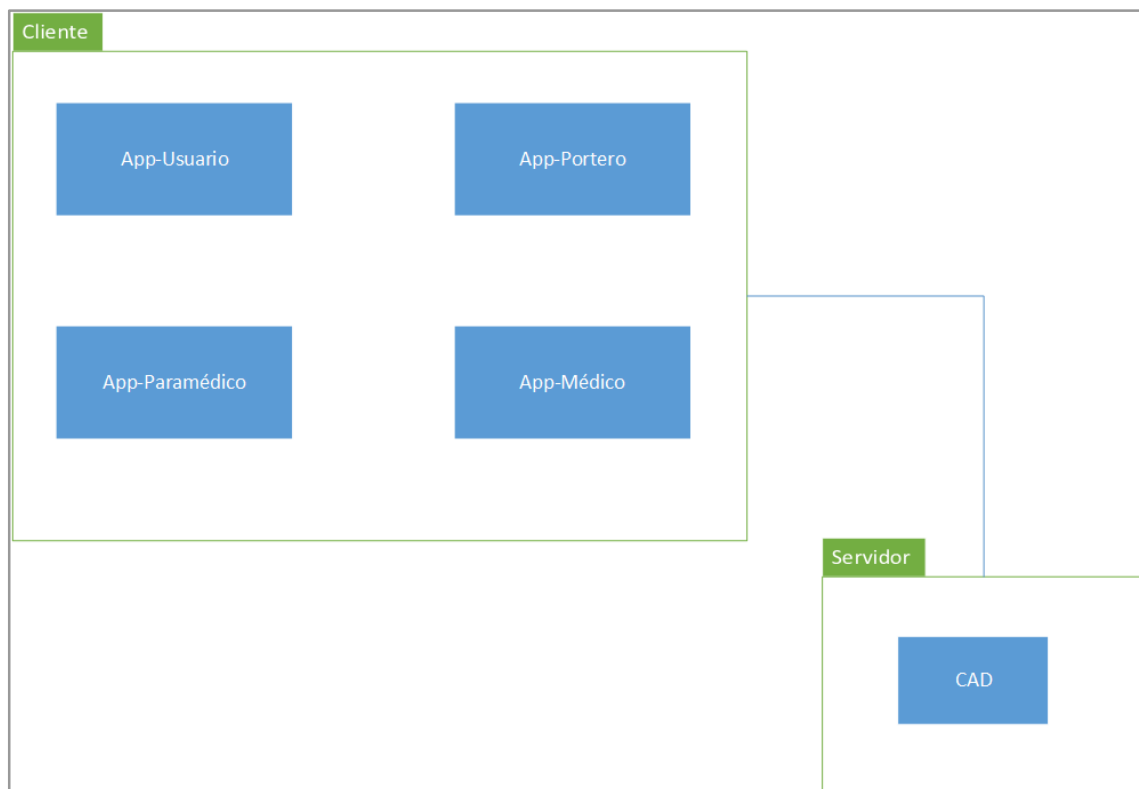


FIGURA 17. MODELO CLIENTE/SERVIDOR

Inicialmente, se describirán las aplicaciones que utilizan los agentes del servicio, es decir, los clientes del sistema.

Para la aplicación del usuario, se definieron las clases de Usuario, Solicitud, Ambulancia y Navegación.

1. **Usuario:** necesaria para entablar comunicación con el CAD y representar lógicamente al usuario.
2. **Solicitud:** encargada de representar la solicitud que se le envía al CAD.
3. **Ambulancia:** tiene como función facilitar el manejo de la información de la ambulancia.
4. **Navegación:** sirve para realizar las labores de navegación, tales como ubicación actual, pintar el mapa geográfico, el camino que toma la ambulancia y el tiempo de llegada de la ambulancia.

Para la aplicación del paramédico se definieron un total de 10 clases diferentes: Paramédico, Navegación, Traslado, Ambulancia, Paciente, Enfermedad, Procedimiento, EPS, IPS y Signos Vitales.

1. **Paramédico:** necesaria para entablar comunicación con el CAD y representar lógicamente al paramédico.
2. **Navegación:** sirve para realizar las labores de navegación, tales como ubicación actual, desplegar el mapa geográfico, lugar de la solicitud de atención, lugar de IPS Destino, tiempo estimado de viaje.
3. **Traslado:** representa lógicamente los traslados realizados por la ambulancia, guardando la información de la fecha de inicio (con hora), fecha de fin (con hora), origen del traslado y destino del traslado, paciente, estado, tipo y toda la información que debe tener un traslado.
4. **Ambulancia:** tiene como función cargar los datos de la ambulancia en la que el paramédico ha iniciado sesión.
5. **Paciente:** representa al paciente llevado en la ambulancia, puede ser un histórico de pacientes, o el paciente que se encuentra en traslado actualmente. El paciente cuenta con la información de sus enfermedades, procedimientos realizados, signos vitales, EPS, entre otros.
6. **Enfermedad:** es la representación de las enfermedades que tiene un paciente. Se tiene un código único y una descripción de la enfermedad.
7. **Procedimiento:** son los procedimientos realizados al paciente. Se tiene un código único y una descripción del procedimiento.
8. **EPS:** es la EPS que tiene el paciente.
9. **IPS:** representa la institución médica de destino del traslado, es decir, a qué institución médica fue llevado el paciente, además, se puede dar el caso de ser una institución de origen del traslado (para el traslado interhospitalario).
10. **Signos Vitales:** son los signos vitales del paciente.

Para la aplicación del médico, se definieron un total de 8 clases: Médico, Traslado, Paciente, Enfermedad, Procedimiento, Signos Vitales, IPS, Navegación.

1. **Médico:** necesaria para entablar comunicación con el CAD y representar lógicamente al médico.

2. **Traslado:** representa lógicamente los traslados atendidos por el médico, guardando la información de la fecha de inicio (con hora), fecha de fin atención (con hora), origen del traslado y destino del traslado, paciente, estado, tipo y toda la información que debe tener un traslado.
3. **Paciente:** representa al paciente atendido por el médico, puede ser un histórico de pacientes, o el paciente que se encuentra en traslado actualmente. El paciente cuenta con la información de sus enfermedades, procedimientos realizados, signos vitales, EPS, entre otros.
4. **Enfermedad:** es la representación de las enfermedades que tiene un paciente. Se tiene un código único y una descripción de la enfermedad.
5. **Procedimiento:** son los procedimientos realizados al paciente. Se tiene un código único y una descripción del procedimiento.
6. **Signos vitales:** son los signos vitales del paciente.
7. **IPS:** representa la institución médica de donde se encuentra ubicado el médico.
8. **Navegación:** sirve para realizar las labores de navegación, tales como ubicación actual, pintar el mapa geográfico, información de la ambulancia, tiempo estimado de viaje de la ambulancia.

Para la aplicación del vigilante, se diseñaron en total 3 clases: Portero, Traslado y Navegación.

1. **Portero:** necesaria para entablar comunicación con el CAD y representar lógicamente al Vigilante del área de urgencias.
2. **Traslado:** representa lógicamente los traslados realizados y programados para la IPS donde el vigilante se encuentra, guardando la información de la fecha de inicio (con hora), fecha de fin atención (con hora), origen del traslado y destino del traslado, paciente, estado, tipo y toda la información que debe tener un traslado.
3. **Navegación:** sirve para realizar las labores de navegación, tales como ubicación actual, pintar el mapa geográfico, información de la ambulancia, tiempo estimado de viaje de la ambulancia.

Finalmente, para el servidor CAD se diseñaron un total de 15 clases: Servidor, Manejador BD, Solicitud, Instituciones, Paramédico, Portero, Ambulancia, Traslado, Médico, Navegación, Paciente, Enfermedad, Procedimiento, Signos Vitales.

1. **Servidor:** necesaria para entablar comunicación con el resto de aplicaciones y representar lógicamente al CAD.
2. **Manejador BD:** utilizado para manejar la base de datos, es la clase que se comunica con la base de datos y se encarga de agregar, buscar, modificar y eliminar un registro, dependiendo de su tipo.
3. **Solicitud:** encargada de representar la solicitud recibida en el CAD.
4. **Instituciones:** son las instituciones que se encuentran activas en el sistema y se encuentran realizando y atendiendo traslados.
5. **Paramédicos:** son los paramédicos que se encuentran activos en el sistema y se encuentran en una ambulancia siendo el responsable de traslados hechos por la ciudad de Cali.

6. **Portero:** son los porteros que se encuentran actualmente activos en el sistema y están atentos a los traslados que se dirigen a la institución.
7. **Ambulancia:** son las ambulancias activas en el sistema y se encuentran disponibles para realizar traslados, sin importar que su estado sea ocupada.
8. **Traslado:** son los traslados activos que se iniciaron y aun no se han finalizado.
9. **Médico:** hace referencia a los médicos que están activos para atender las emergencias que llegan a los diferentes centros de salud.
10. **Navegación:** sirve para realizar las labores de navegación, tales como pintar el mapa geográfico, ubicación de los diferentes agentes.
11. **Paciente:** Son los pacientes que se encuentran activos en el sistema.
12. **Enfermedad:** es la representación de las enfermedades que tiene un paciente activo. Se tiene un código único y una descripción de la enfermedad.
13. **Procedimiento:** son los procedimientos realizados al paciente activo. Se tiene un código único y una descripción del procedimiento.
14. **Signos Vitales:** son los signos vitales del paciente activo.

Este entregable es la guía para iniciar la implementación del SIEF, permite el trabajo en conjunto de los desarrolladores del servicio, por lo que es de suprema importancia. Para revisar todos los diagramas de clases realizados, puede dirigirse al anexo "*DiagramaClases.vsd*".

7.8. DESARROLLO DEL SISTEMA

En esta sección, se describirán todos los procesos realizados para desarrollar un sistema que cumpla lo pactado en la sección de análisis y diseño, teniendo en cuenta requerimientos funcionales, requerimientos no funcionales, esquema funcional y técnico. Además, siguiendo el mismo lineamiento, se hizo un desarrollo para un componente de FrontEnd -Aplicaciones móviles y web para los usuarios finales- y un componente de BackEnd -Servidor con manejo de hilo y diferentes transacciones; y una base de datos relacional.

Inicialmente, se describe el componente de FrontEnd, explicando el proceso de desarrollo, empezando por la preparación del ambiente de desarrollo de las diferentes aplicaciones móviles y web. Por parte del componente de BackEnd, explicando los procedimientos para preparar el ambiente de desarrollo, la implementación de la base de datos y los diferentes WebServices encargados de manejar las solicitudes de los usuarios.

Un diagrama general explicativo de los componente a desarrollar es el siguiente:

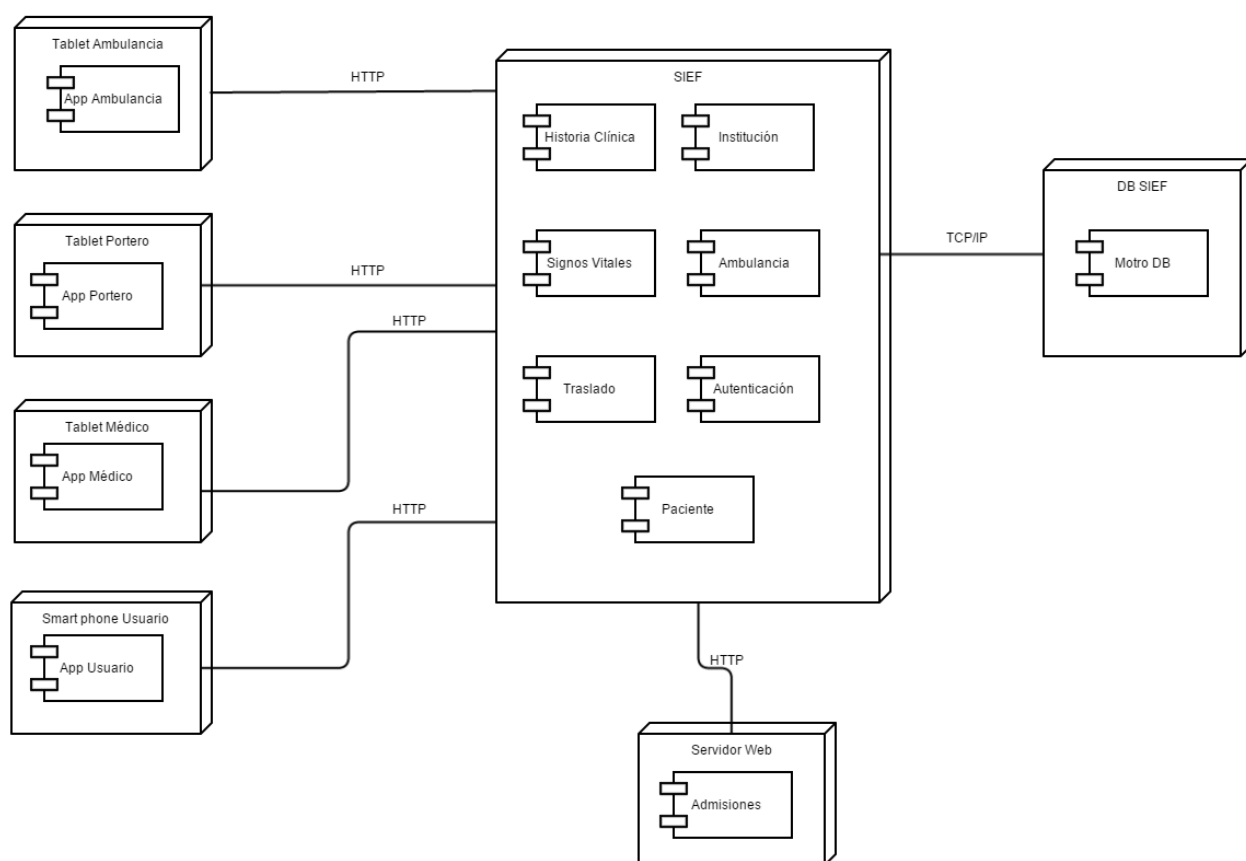


FIGURA 18 DIAGRAMA GENERAL COMPONENTES SIEF

7.9. COMPONENTE DE FRONT END

7.9.1 SOFTWARE DE DESARROLLADO INSTALADO

Para el desarrollo de la funcionalidad se utilizaron diferentes componentes de software para la arquitectura de la solución los cuales son:

- **Android Studio:** Un entorno de desarrollo integrado (IDE) para la plataforma Android. Fue anunciado por Ellie Powers el 16 de mayo de 2013. Android Studio está disponible para desarrolladores para probarlo gratuitamente. Basado en IntelliJ IDEA de JetBrains, está diseñado específicamente para desarrollar para Android. Está disponible para descargar para Windows, Mac OS X y Linux (Google Inc, 2015).

Android Studio nos ofrece las siguientes ventajas:

- Renderización en tiempo real
- Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.
- Soporte para construcción basada en Gradle.
- Refactorización específica de Android y arreglos rápidos.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones, y otros problemas.
- Plantillas para crear diseños comunes de Android y otros componentes.
- Soporte para programar aplicaciones para Android Wear.

Esta fue la herramienta que permitió el desarrollo de las aplicaciones móviles, por otro lado, la aplicación web se realizó utilizando el entorno de desarrollo Eclipse, ya que permite el desarrollo utilizando lenguaje html, código jsp y manejo de servlets. La descripción del desarrollo de la aplicación web será realizada en la sección para el componente de Back-End, debido a que su desarrollo va de la mano con la implementación del servidor en google cloud y éste trabajó como un proyecto en conjunto, aunque tenga una interacción directa con el usuario final. De esta manera, sólo se mostrará en esta sección el desarrollo de las aplicaciones móviles.

7.9.2 PREPARACIÓN DE AMBIENTE DE DESARROLLO

En esta sección, se describe el proceso necesario de preparación de un ambiente de desarrollo que permitiera implementar el servicio SIEF en sus componentes de Front End.

- **Android Studio** se conoce como la herramienta de desarrollo del futuro, pues en poco tiempo será lo único que el equipo de Android recomiende. El desarrollo en Android Studio es más versátil, más potente, actual y mantiene una similitud con eclipse java, se pueden cargar distintos recursos, apuntar a varios servicios web lo que facilita el desarrollo de SIEF.

El proceso de instalación y preparación del ambiente Android Studio para el desarrollo consta de los siguientes pasos. Ver diagrama de Flujo en la figura 18.

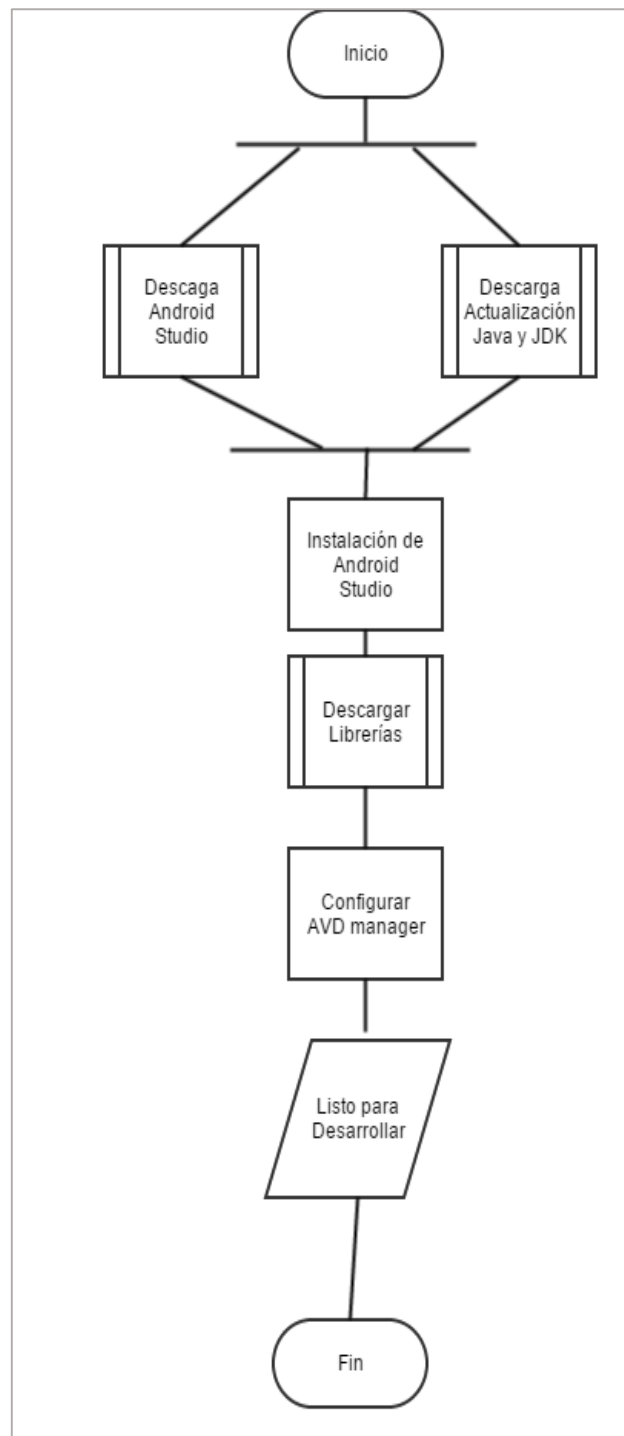


FIGURA 19. DIAGRAMA DE FLUJO ANDROID STUDIO

7.10. IMPLEMENTACIÓN APLICACIONES MÓVILES

Una vez listo el ambiente de desarrollo, con el Android Studio instalado y actualizado, se inicia el proceso de implementación de cada una de las aplicaciones teniendo en cuenta los diseños presentados previamente para lograr cumplir con los requerimientos funcionales planteados.

7.10.1 APLICACIÓN USUARIO

Esta aplicación se desarrolló en tres etapas, en cada una de ellas se realizaron pruebas de funcionalidad para verificar el correcto funcionamiento de la implementación de la aplicación. La primera fase fue el desarrollo de la disposición de los componentes gráficos, para tener una interfaz gráfica de usuario que cumpliera con lo estipulado en las maquetas, realizando las mejoras necesarias para tener una interfaz más amigable y eficiente en la interacción con el usuario. Para hacerlo se tuvieron en cuenta los patrones de diseño de aplicaciones móviles y retroalimentación de usuarios tras la experiencia al presentar el modelo inicial, para así cumplir con uno de los requerimientos de funcionalidad de la aplicación. La segunda fase de desarrollo, se centró en la implementación de las diferentes clases estipuladas en el diagrama de clases, que permiten tener un manejo de la información ordenado y así tener una aplicación más estable en funcionamiento; la última fase se centralizó en establecer comunicación con el servidor y hacer el intercambio de información entre las partes.

En la primera fase, se hizo un desarrollo de la disposición de los componentes gráficos de la aplicación móvil utilizando la herramienta Android Studio. Esta parte visual de la aplicación es de crucial importancia porque el usuario debe comprender fácilmente el funcionamiento de la aplicación y en pocos pasos debe poder solicitar una ambulancia para la atención de su emergencia. Además, la información que muestra la aplicación debe presentarse de una manera en que su lectura sea intuitiva. Se hicieron un total de 4 ventanas principales de interacción con el usuario, cada una cumpliendo con los objetivos pactados de funcionalidad e interacción, permitiendo al usuario: reconocimiento de la marca, solicitar la atención de una emergencia en su lugar de ubicación actual, además de la opción de modificar la dirección de origen de la emergencia (visualización de ubicación en mapa), visualizar la información de la ambulancia que va en camino (incluyendo el tiempo estimado de llegada al lugar de la emergencia), e identificar las instituciones médicas cercanas, se utilizan dos ventanas emergentes una para aviso de búsqueda de ambulancia y otra para el manejo de la calificar la aplicación. Durante el desarrollo, se realizaron validaciones en el emulador de un celular con sistema operativo Android. El proceso de desarrollo total de esta aplicación tuvo un tiempo aproximado de 2 semanas.

La sección de manejo de mapas es de vital importancia, por lo que se decidió trabajar el sistema de navegación utilizando las librerías de google maps, el cual permite la visualización del mapa señalando la ubicación actual o indicada. Cuando se trata de la ubicación actual, se utiliza el GPS del dispositivo. En el mapa se puede visualizar la ambulancia que se encuentra en camino y las instituciones médicas más cercanas. Para el desarrollo de esta parte del proceso, fue necesario agregar la librería de google maps al proyecto, entender su funcionamiento de parámetros y resultados para obtener los resultados visuales (posición de un dispositivo móvil, longitud, latitud). Debido a que era necesario utilizar un GPS, se hizo necesario el uso de un equipo físico para realizar las pruebas, lo que implicó la configuración del Android Studio a la hora de ejecutar

la aplicación móvil, permitiendo la ejecutará con un archivo .apk el cual es ejecutado en un dispositivo móvil. Este dispositivo debe ser conectado por un cable USB al computador donde estaba el Android Studio, además, se debe activar las funciones de desarrollador en el Smart Phone pulsando repetidamente la opción “Número de compilación” en el panel de configuraciones. El proceso para tener funcionando las características de navegación tomo un tiempo aproximado de 1 semana.

La segunda fase se desarrolló utilizando los diagramas de clases de las representaciones de los agentes que interactúan con el usuario final, para contar con una aplicación estable y funcional que permite manejar la información, que será presentada al usuario, de manera ordenada para poder cargar los datos de una mejor manera en la aplicación móvil. Para lograr esto se utilizan objetos, que representan al usuario, la ambulancia y los demás objetos necesarios para representar las partes del SIEF. Para ver con mayor detalle el objetivo de cada una de estas clases el lector debe dirigirse a la sección de diagramas de clases. El desarrollo de esta etapa se realizó con el lenguaje de programación Java, debido a que es el utilizado por las aplicaciones Android y se tiene conocimiento previo del mismo porque es utilizado en los cursos de programación de la Universidad Icesi. Este desarrollo tomo un tiempo aproximado de 1/2 semana.

Una vez se tiene la estructura necesaria para representar lógicamente a los usuarios, solo falta realizar el desarrollo necesario para establecer comunicación con el servidor. Este desarrollo se basa en los diagramas de mensajes e intercambio de etiquetas XML establecidos para conseguir demandar exitosamente cada uno de los servicios web necesarios para el correcto funcionamiento de esta aplicación. Primero se deben configurar los parámetros utilizados para establecer comunicación con el servidor, estos son la dirección IP, el nombre del servicio web que atenderá la solicitud, el método a utilizar y la solicitud (request) que debe enviarse. Con las respuestas obtenidas se logra: actualizar en tiempo real la ubicación de la ambulancia, ofreciendo así una aplicación completa que cumple con los requerimientos funcionales. Los WebServices fueron implementados usando una estructura tipo SOAP que permite encapsular etiquetas XML dentro de una solicitud Http, así, las transmisiones de datos podrán ser enviadas al servidor en una sola transacción sin riesgo a que se perdiera información por cada mensaje enviado y recibido. Esta implementación tomó un tiempo aproximado de 2 semanas.

La aplicación final de usuario se puede apreciar a continuación:

Reconocimiento de Logotipo. Figura 19.



FIGURA 20. RECONOCIMIENTO DE MARCA

Pantalla principal de solicitud de servicio, vista como la figura 20 y 21.



FIGURA 21. CARGANDO POSICIÓN ACTUAL.



FIGURA 22. POSICIÓN ACTUAL ENCONTRADA.

Al seleccionar cualquiera de las opciones del tipo de emergencias se carga la ventana emergente de búsqueda de ambulancia. Figura 22.

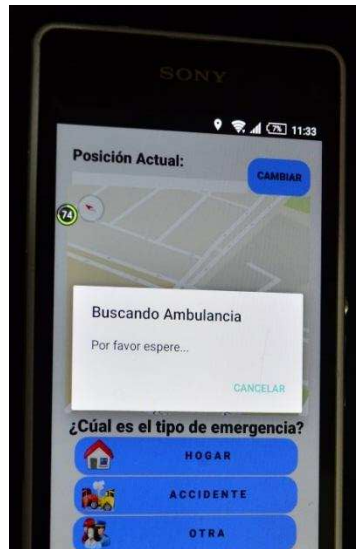


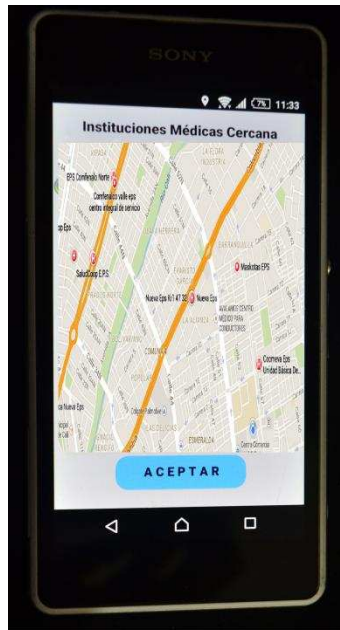
FIGURA 23. BUSQUEDA DE AMBULANCIA

Una vez encontrada la ambulancia, se muestra la información relevante de la ambulancia, la ruta y el tiempo aproximado de llegada, además de un botón que permite al usuario indicar que la ambulancia llegó a prestar la atención. Ver figura 23.



FIGURA 24. RUTA ENTRE AMBULANCIA Y UBICACIÓN INICIAL

Cuando el usuario y/o la ambulancia notifican al CAD que se encuentra en el sitio del llamado se muestran las Instituciones Médicas más cercanas, figura 24.



La última pantalla de la aplicación corresponde con el mensaje emergente que permite calificar el servicio, la apreciación visual se puede observar en las figura 25 y26.

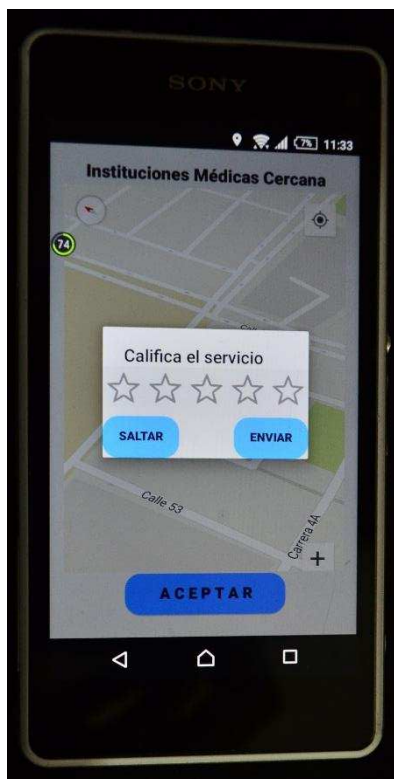


FIGURA 27. CALIFICACIÓN /SALTAR/ENVIAR



FIGURA 26. CALIFICAR

Con esto se daría por concluido el servicio de solicitud de traslado por parte de un usuario de Smart Phone.

7.10.2 APLICACIÓN PARAMÉDICO

Esta aplicación se desarrolló igual que la aplicación de usuario, dividiendo la implementación en 3 fases principales: desarrollo de las vistas principales con las funciones de navegación y notificaciones, desarrollo de las diferentes clases necesarias para generar la abstracción lógica de los usuarios, por último el desarrollo de la comunicación con el servidor y poder intercambiar información con las demás partes.

La primera fase se centró en el desarrollo de una disposición gráfica para la aplicación móvil del paramédico utilizando la herramienta Android Studio. Esta parte visual de la aplicación es de vital importancia ya que el paramédico debe comprender fácilmente el funcionamiento de la aplicación y en pocos pasos debe poder manejar la información de su perfil, el historial del traslado de la ambulancia, cambiar disponibilidad, atender una solicitud, documentar un paciente e intercambiar información con la IPS destino principalmente, esta información se presenta en la aplicación de una manera en que su lectura sea fácil e intuitiva. Se hicieron un total de 10 ventanas principales de interacción con el paramédico o médico responsable de la ambulancia, cada una cumpliendo con los objetivos pactados de funcionalidad e interacción, permitiendo al paramédico cumplir con sus tareas. Estas vistas se basan en mostrar de una manera amigable una pantalla de acceso al sistema, funcionalidades de la aplicación, como historial de traslados, cambio de disponibilidad y contraseña, atención de pacientes, además, de mostrar los mapas necesarios para realizar los traslados desde un lugar de emergencia y documentar un paciente actualizando los signos vitales. Adicionalmente, se tienen una serie de notificaciones que deben ser claras en su lectura e interpretación, por lo que tienen mensajes cortos y claros. La simulación se realizó en una tablet de 10" el tiempo aproximado de desarrollo fue de 2 semanas.

En la etapa de funciones de navegación fue necesario realizar los mismos procesos de configuración descritos en la aplicación del usuario. El desarrollo requerido para tener funcionando las características de navegación en la aplicación del paramédico tomó un tiempo aproximado de 1 semana.

La segunda fase se desarrolló utilizando los diagramas de clases de las representaciones de los agentes que interactúan con los paramédicos o el médico responsable de la ambulancia, para poder tener una aplicación estable y funcional, que permitiera manejar la información, de manera ordenada para poder cargar los datos de una mejor manera en la aplicación móvil. Para el desarrollo de esta aplicación, resulta supremamente importante implementar correctamente las clases que definen los objetos y su comportamiento en el sistema, es decir, con las clases se pueden establecer las operaciones que puede realizar un paramédico a cargo de la ambulancia, definir sus atributos y los métodos a utilizar para que el sistema cumpla con los requerimientos de este agente. Con esta aplicación el personal de la ambulancia puede manejar los traslados y los pacientes, de una manera eficiente debido a que es posible cargar en la memoria RAM del dispositivo el objeto que representa el traslado actual que se está realizando, mejorando los tiempos de respuesta; así no será necesario enviar toda la información generada por un traslado inmediatamente al servidor, sino que se utiliza el CAD solo para el intercambio de información crítica mientras el traslado está en camino, para manejar localmente este objeto. Una vez se

culmina el traslado, la aplicación envía todos los datos al servidor para que estos sean registrados en la base de datos. Se tomó la decisión de implementar esta fase de esta manera, para mejorar la eficiencia en los tiempos de respuesta de las interacciones del usuario con la aplicación. Esta etapa se implementó utilizando Java y su desarrollo tomó un tiempo aproximado de 2 semanas.

Por último, la comunicación con el servidor se realiza igual que con la aplicación de usuarios, primero se deben configurar los parámetros utilizados para establecer comunicación con el servidor, dichos parámetros son el espacio de nombre, que ayuda establecer donde se aloja el servicio web, el nombre del método demandado y la estructura para definir el mensaje que se envía, incluyendo el esquema XML y la solicitud tipo HTTP. Con las respuestas obtenidas se logra: intercambiar información con los demás agentes, actualizar en tiempo real la ubicación de la ambulancia, enviar información del estado del paciente a la institución médica, entre otras; ofreciendo así una aplicación completa que cumple con los requerimientos funcionales. La implementación tomó un tiempo aproximado de 2 semanas.

La aplicación del paramédico como se puede observar en las figura 27 hasta la 39.

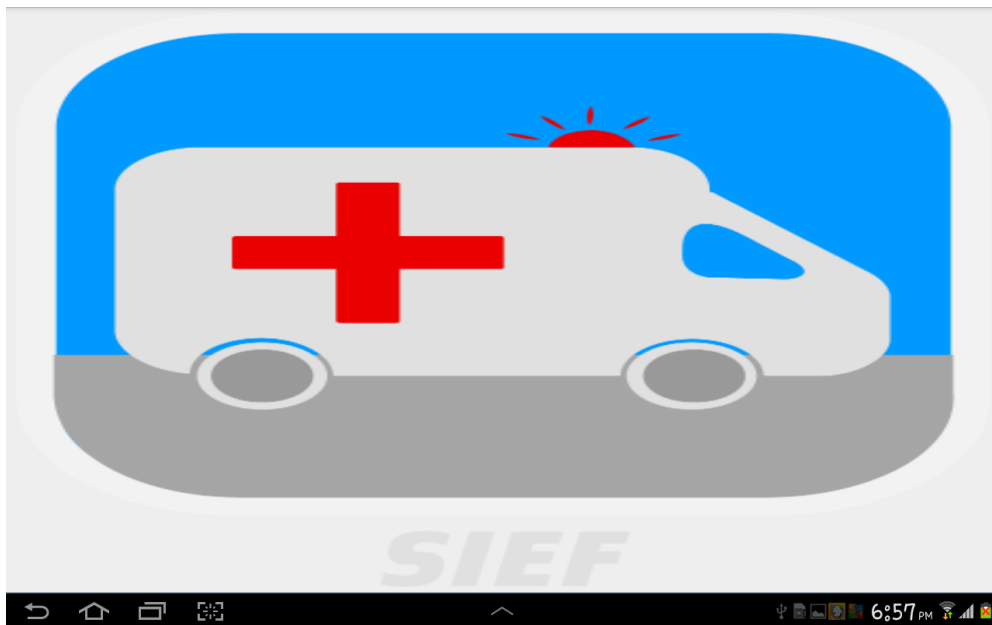


FIGURA 28. CARGUE INICIAL, RECONOCIMIENTO DE MARCA



FIGURA 29. PANTALLA LOGIN



FIGURA 30. PANTALLA DE BIENVENIDA



FIGURA 31. CAMBIO DE ESTADO.



FIGURA 32. MOMENTO DE EMERGENCIA O SOLICITUD DE SERVICIO

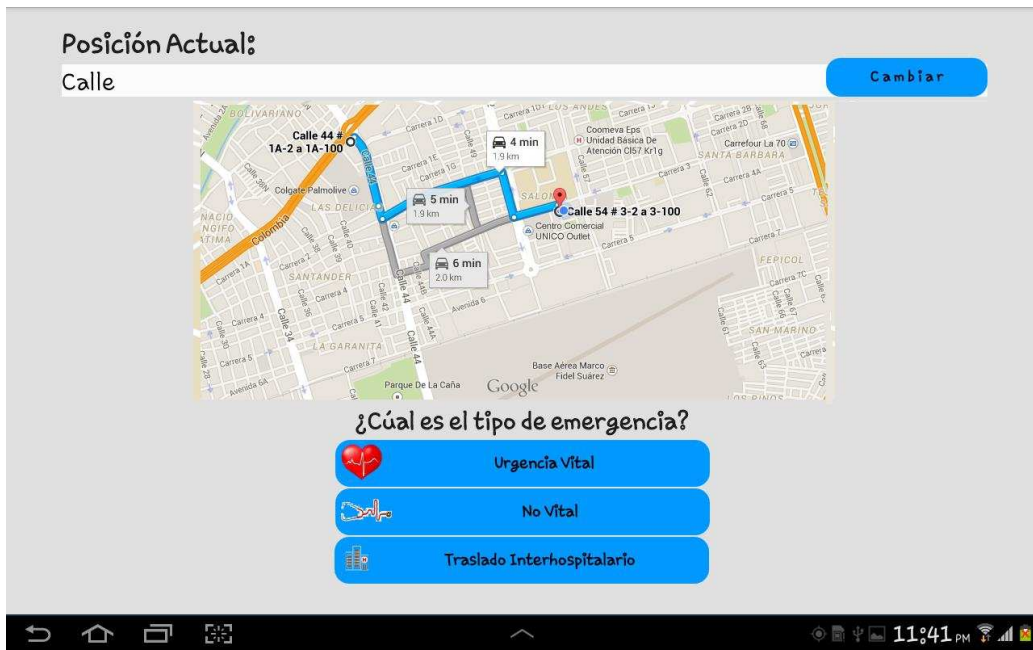


FIGURA 33. PANTALLA DE NUEVO TRASLADO

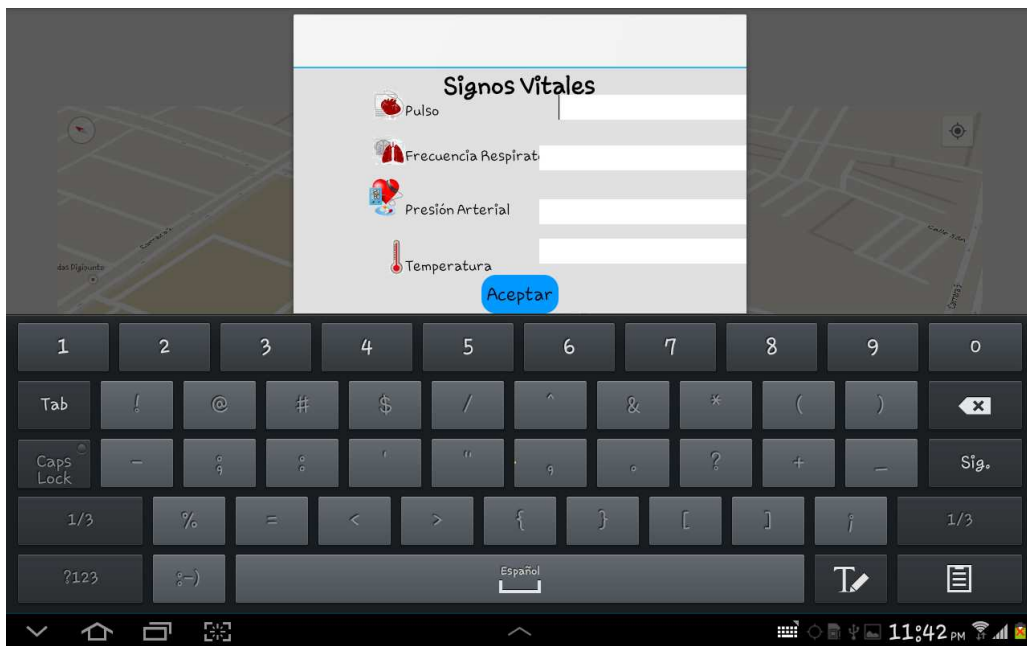


FIGURA 34. PANTALLA INGRESO DE SIGNOS VITALES

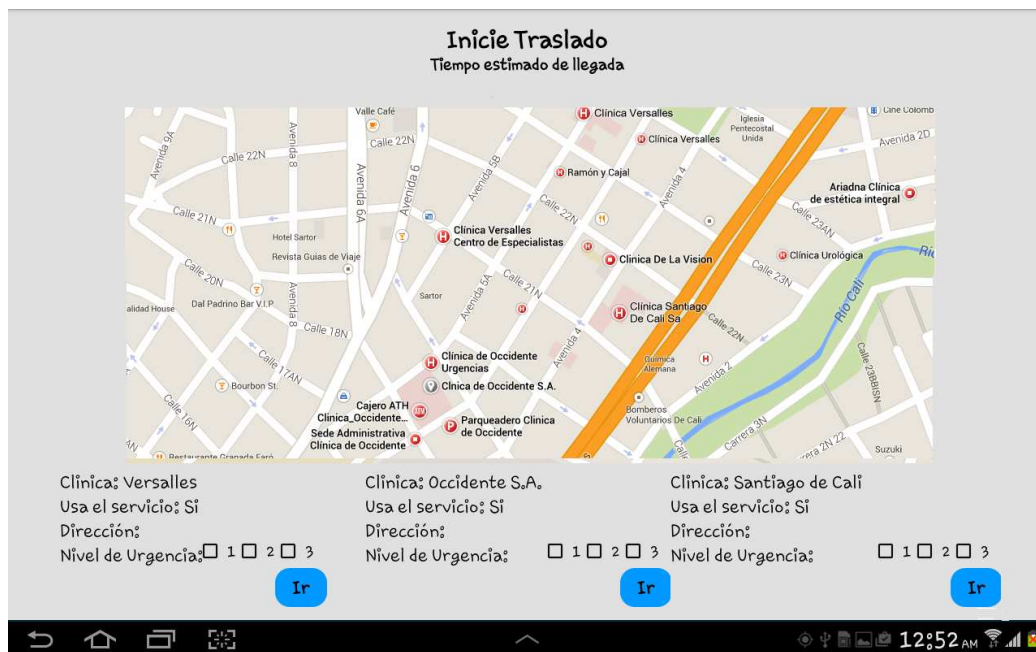


FIGURA 35. SELECCIONAR UNA INSTITUCIÓN CERCANA

Información del Paciente

Datos Básicos:

Nombres: Apellidos:
 Tipo de documento: ☐ RG ☐ TI ☐ C.C.
 Número:
 Edad: Sexo: ☐ F ☐ M

Información EPS

EPS: ☐ Cotizante ☐ Beneficiario
 Nivel: ☐ A ☐ B ☐ C

Información Traslado

Origen:
 Destino:

Información Clínica

Nivel de urgencia: ☐ Vital ☐ No Vital Otro: Alergias:
 Procedimientos Realizados:
 Resumen de urgencia:

Signos Vitales **Solicitar Atención**

11:42 PM

FIGURA 36. PANTALLA INGRESO DE INFORMACIÓN DEL PACIENTE

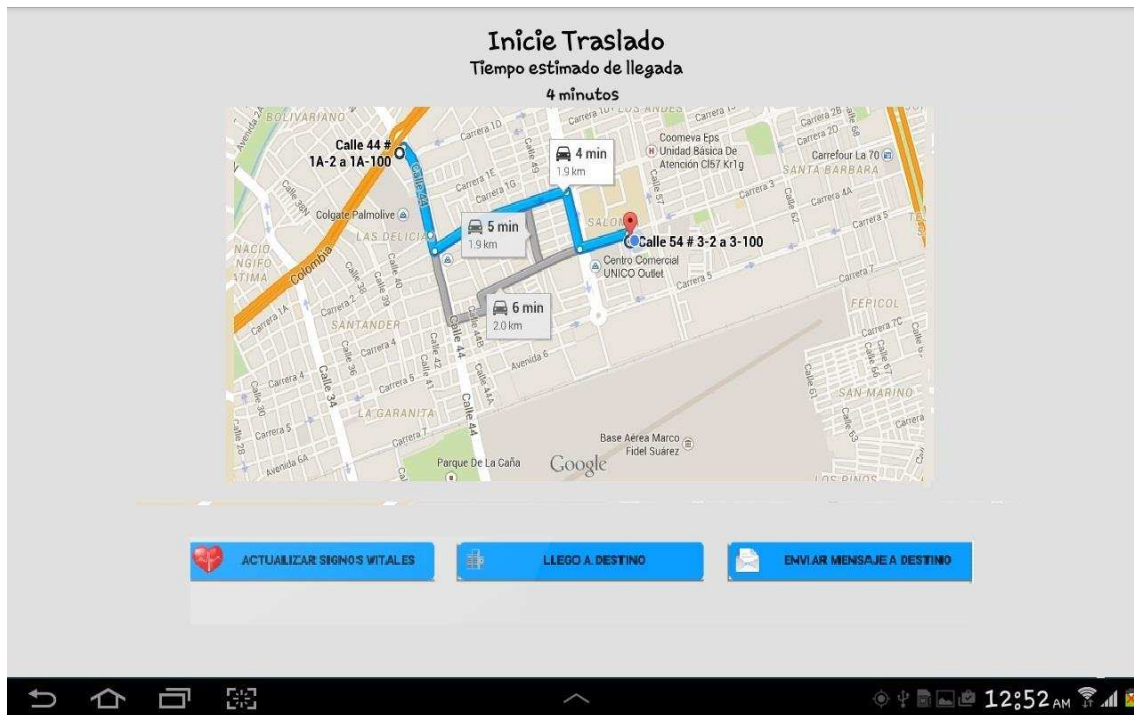


FIGURA 37. TRASLADO EN CURSO

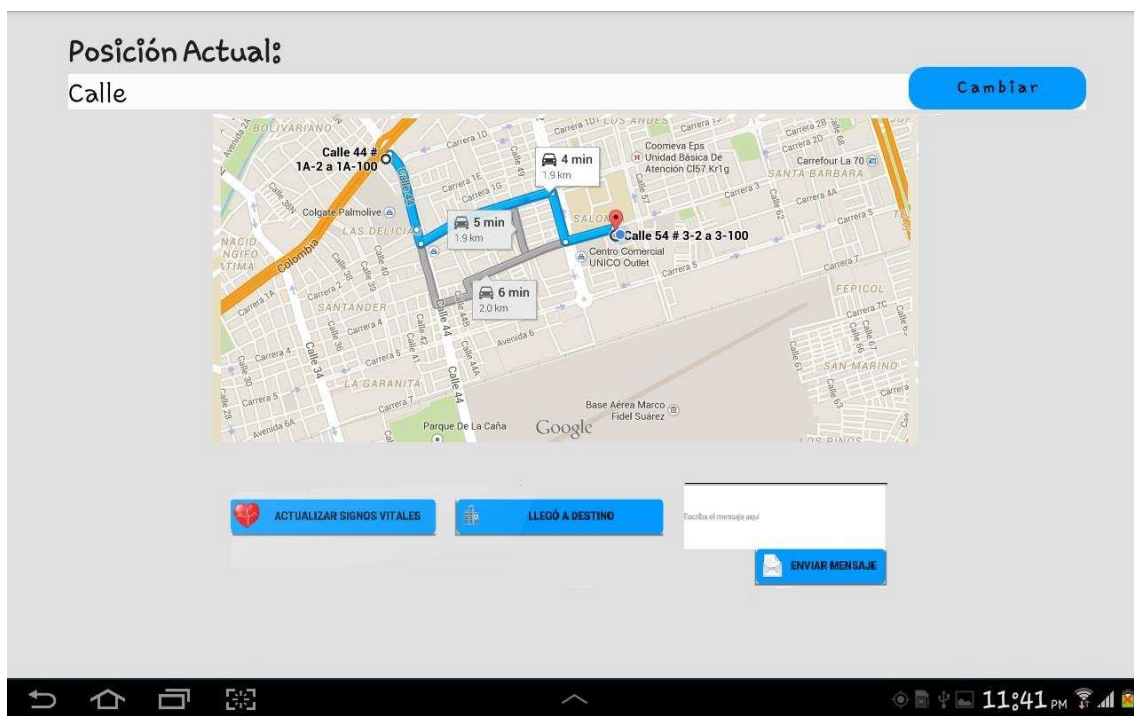


FIGURA 38. ENVÍO DE MENSAJE EN EL CAMINO

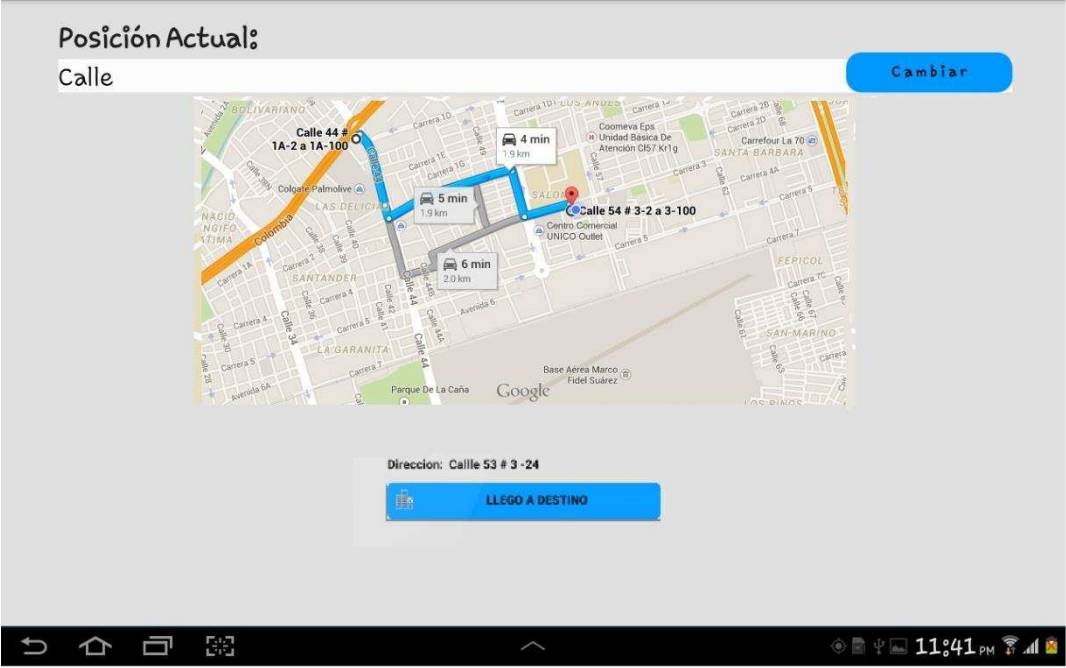


FIGURA 39. TRASLADO INTERHOSPITALARIO

Servicios Atendidos						
#Servicio	Fecha	Hora Inicio	Hora Fin	Origen	Destino	Información Paciente
1	27/03/2015	8:35	12:35	Calle 54 con 3ra	Comfenalco	Ver Paciente
1	27/03/2015	8:35	12:35	Calle 54 con 3ra	Comfenalco	Ver Paciente
1	27/03/2015	8:35	12:35	Calle 54 con 3ra	Comfenalco	Ver Paciente
1	27/03/2015	8:35	12:35	Calle 54 con 3ra	Comfenalco	Ver Paciente
1	27/03/2015	8:35	12:35	Calle 54 con 3ra	Comfenalco	Ver Paciente
1	27/03/2015	8:35	12:35	Calle 54 con 3ra	Comfenalco	Ver Paciente

Volver

11:35 PM

FIGURA 40. LISTA DE SERVICIOS ATENDIDOS.

7.10.3 APLICACIÓN VIGILANTE

Esta aplicación se desarrolló siguiendo la misma estructura de las aplicaciones anteriores, dividiendo la implementación en 3 fases principales: desarrollo de las vistas principales con las funciones de navegación y notificaciones, desarrollo de las diferentes clases necesarias para generar la abstracción lógica de los usuarios, por último el desarrollo de la comunicación con el servidor y poder intercambiar información con las demás partes.

La primera fase se centró, nuevamente, en el desarrollo de la interfaz gráfica para la aplicación móvil, pero en este caso para la aplicación de vigilante, utilizando la misma herramienta (Android Studio). Esta parte visual tiene modificaciones a partir de la interfaz visual de la aplicación de ambulancia. El vigilante debe comprender fácilmente el listado de traslados en camino y los que se encuentran en espera, visualizar la información del traslado y entender la urgencia del traslado, cabe aclarar, que el portero no da una validación del estado del paciente, pero es importante que sepa el nivel de urgencia para poder anticipar y ayudar con el ingreso del paciente a la institución. Esta información se presenta en la aplicación de una manera en que su lectura sea fácil e intuitiva. Se hicieron un total de 4 ventanas principales de interacción con el portero, cada una orientada a ayudar al portero en su labor del ingreso a la unidad de urgencias, sin perder funcionalidad e interacción, permitiendo al vigilante cumplir con sus tareas. Estas vistas se basan en mostrar de una manera amigable una pantalla de acceso al sistema, las diferentes opciones de la aplicación, como un listado de traslados programados y en camino a la institución y la información del traslado. Adicionalmente, se tienen una serie de notificaciones que deben ser claras en su lectura e interpretación, por lo que tienen mensajes cortos y claros. Esta fase se desarrolla utilizando la herramienta Android Studio y se verificaron las ventanas con el emulador de una tablet de 10" con sistema operativo Android y tuvo un tiempo aproximado de 1 semana.

La fase de navegación conserva el mismo tipo de desarrollo y configuración de la aplicación del paramédico, permitiendo la visualización de un mapa donde puede ver la ambulancia que se encuentra en camino, el tipo de emergencia y dato relevantes del paciente según sea el caso, para gestionar su ingreso. Para el desarrollo de esta parte fue necesario realizar los mismos procesos de configuración descritos en la aplicación del usuario. El proceso para tener funcionando las características de navegación en la aplicación del paramédico tomó un tiempo aproximado de 1/2 semana.

La segunda fase tiene el mismo desarrollo y modelo de las anteriores, el desarrollo tomó un tiempo aproximado de 1/3 semana.

La comunicación con el servidor conserva la misma estructura planteada para la aplicación del paramédico, la variación consta de las respuestas obtenidas, esta app recibir datos (información del traslado, ubicación, tiempo estimado) desde la ambulancia, envío de notificación del ingreso del paciente; el tiempo aproximado de 2 semanas.

Las vistas de la aplicación del portero se pueden observar desde la figura 40 hasta la 45.



FIGURA 41. RECONOIMIENTO DE MARCA

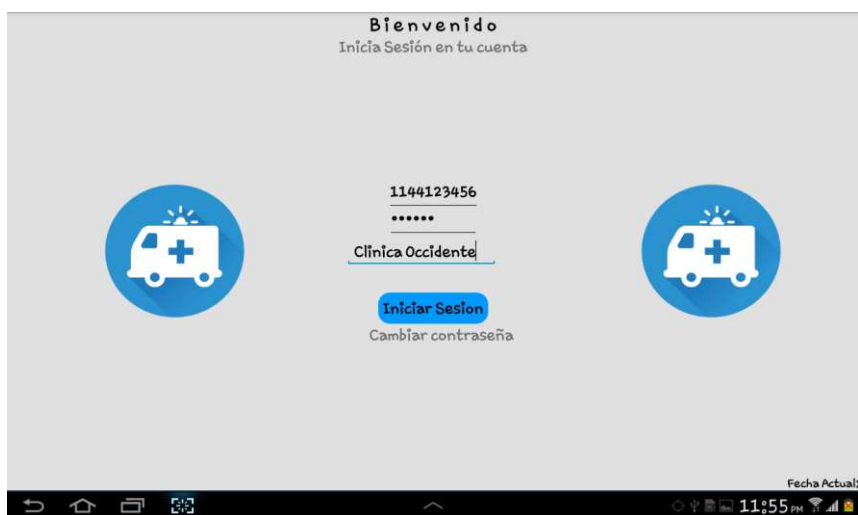


FIGURA 42. PANTALLA DE LOGIN



FIGURA 43. MENÚ DE BIENVENIDA



FIGURA 44. MENSAJE DE EMERGENCIA EN CAMINO

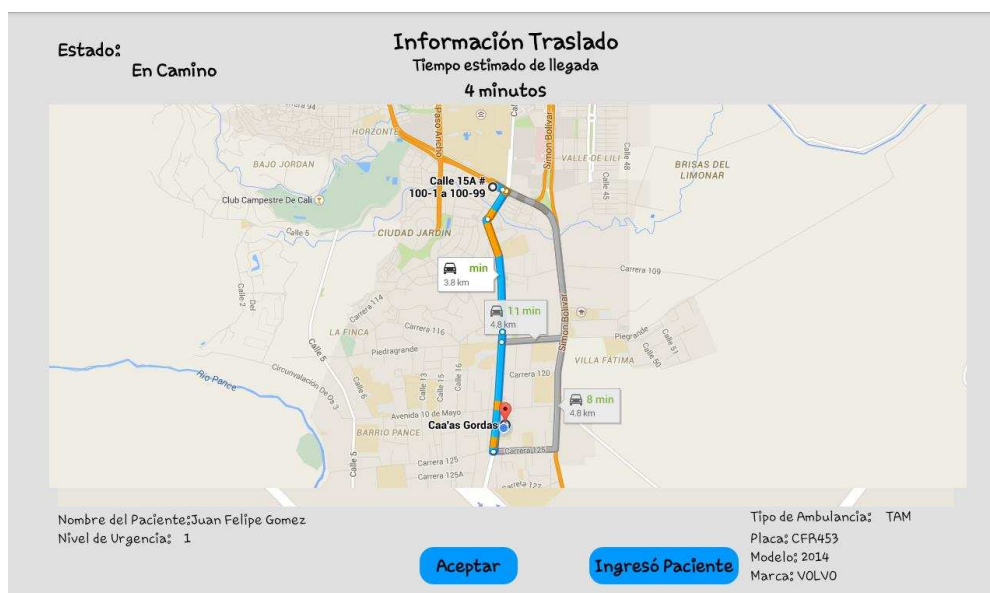


FIGURA 45. PANTALLA INFORMACIÓN DEL TRASLADO

Confirmar Ingreso	Paciente	Estado	Resumen de Urgencia	Tipo de Urgencia	Responsable	Acompañante	Traslado
<input checked="" type="checkbox"/>	Luz Angela Sarria	En Camino	3	SOAT	Comfenalco	Amana Quintero	Ver Paciente
<input checked="" type="checkbox"/>	Luz Angela Sarria	En Camino	3	SOAT	Comfenalco	Amana Quintero	Ver Paciente
<input checked="" type="checkbox"/>	Luz Angela Sarria	En Camino	3	SOAT	Comfenalco	Amana Quintero	Ver Paciente
<input type="checkbox"/>	Luz Angela Sarria	En Camino	3	SOAT	Comfenalco	Amana Quintero	Ver Paciente
<input type="checkbox"/>	Luz Angela Sarria	En Camino	3	SOAT	Comfenalco	Amana Quintero	Ver Paciente
<input type="checkbox"/>	Luz Angela Sarria	En Camino	3	SOAT	Comfenalco	Amana Quintero	Ver Paciente
<input type="checkbox"/>	Luz Angela Sarria	En Camino	3	SOAT	Comfenalco	Amana Quintero	Ver Paciente

Volver

FIGURA 46. LISTA DE TRASLADOS A CONFIRMAR

7.10.4 APLICACIÓN MÉDICO

Esta aplicación se desarrolló siguiendo la misma estructura de las aplicaciones anteriores, dividiendo la implementación en 3 fases principales: desarrollo de las vistas principales con las funciones de navegación y notificaciones, desarrollo de las diferentes clases necesarias para generar la abstracción lógica de los usuarios, por último el desarrollo de la comunicación con el servidor y poder intercambiar información con las demás partes.

La primera fase se centró, nuevamente, en el desarrollo de la disposición de los componentes gráficos para la aplicación móvil. La parte visual de la aplicación permite al médico observar fácilmente el listado de traslados en camino y la información de los pacientes, para conocer su estado de salud mientras se realiza el traslado, teniendo en cuenta el nivel de urgencia, los signos vitales, enfermedades diagnosticadas y procedimientos realizados. Esta información se presenta en la aplicación de una manera en que su lectura sea fácil e intuitiva. Se hicieron un total de 7 ventanas principales de interacción con el médico encargado del área de urgencias, cada una orientada a ayudar al médico en su labor de brindar la mejor atención a los pacientes y poder asegurar el bienestar de sus pacientes y hacer todo lo posible para salvar vidas, ya que al conocer el estado del paciente es posible planificar un procedimiento que logre curar al paciente que está siendo trasladado. Estas vistas muestran de una manera amigable una pantalla de acceso al sistema y las diferentes opciones de la aplicación como: un listado de traslados programados y en camino a la institución, cambio de contraseña, la información del traslado, la información del paciente y al final de la atención, documentar al paciente para finalizar el registro del paciente en la unidad de urgencias. Adicionalmente, se tienen una serie de notificaciones que deben ser claras en su lectura e interpretación, por lo que tienen mensajes cortos y claros. Esta fase se desarrolla utilizando la herramienta Android Studio y se verificaron las ventanas con el emulador de tablet y tuvo un tiempo aproximado de 2 semanas.

La fase de navegación conserva el mismo tipo de implementación y configuración de la aplicación del paramédico. El proceso para tener funcionando las características de navegación en la aplicación del paramédico tomó un tiempo aproximado de 1/2 semana.

La segunda fase se desarrolló utilizando los diagramas de clases de las representaciones de los agentes que interactúan con el médico, para contar con una aplicación estable y funcional que permita manejar la información de los traslados y pacientes atendidos por el urgenciólogo en la unidad de urgencias de la institución médica, que será presentada al usuario, en este caso el médico, de manera ordenada para poder cargar los datos de una mejor manera en la aplicación móvil. Para lograr esto se utilizan objetos, que representan al paciente, la ambulancia, el médico y demás objetos necesarios para la representación de los agentes del SIEF. Gracias a esta implementación es posible manejar la información de los traslados, pacientes (signos vitales, enfermedades y procedimientos), solicitudes y navegación de una manera eficaz. El desarrollo tomó un tiempo aproximado de 1/3 semana.

La comunicación con el servidor conserva la misma estructura planteada en las aplicaciones descritas previamente, el cambio se presenta en los datos que se reciben desde la ambulancia (información del traslado, ubicación, tiempo estimado), el envío de mensajes al servidor, la actualizar en tiempo real los traslados con destino a la IPS, al actualizar en tiempo real los signos vitales del paciente, envío de notificación de la atención del paciente, entre otras; completando los requerimientos funcionales descritos anteriormente. Estos Servicios Web funcionan de igual manera, utilizando la estructura tipo SOAP y su implementación tomó un tiempo aproximado de 2 semanas.

El resultado final y visual de la aplicación del médico se encuentra en las figuras 46 a 51:



FIGURA 47. PANTALLA INICIAL DE MARCA

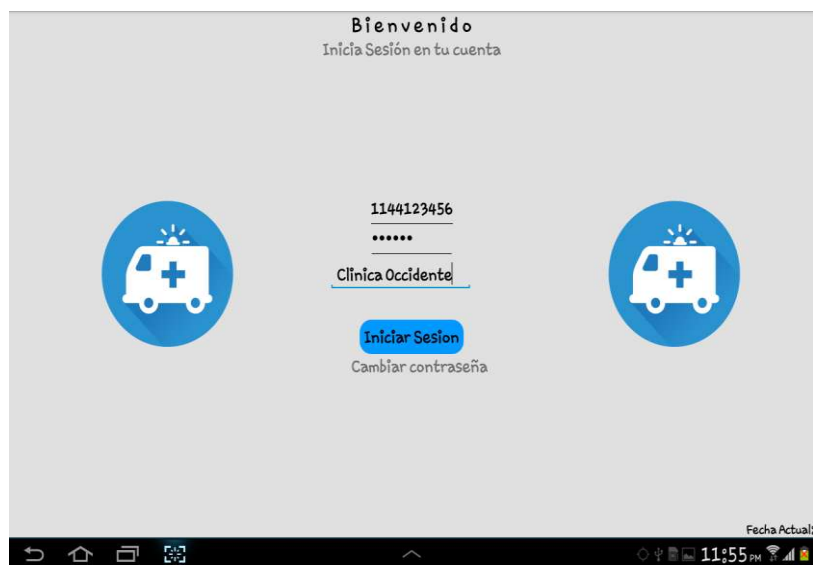


FIGURA 48. PANTALLA DE LOGIN

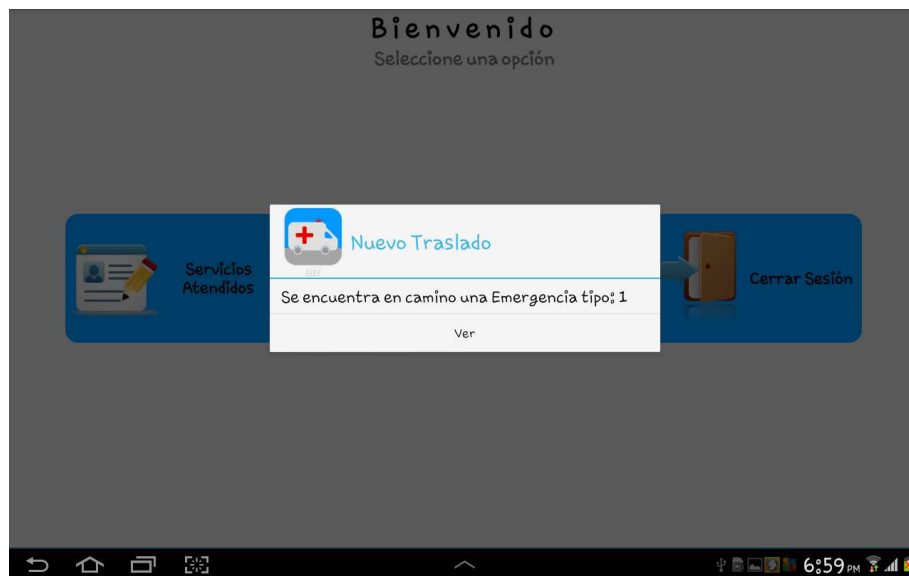


FIGURA 49. MENSAJE NUEVA EMERGENCIA



FIGURA 50. PANTALLA INFORMACIÓN DEL TRASLADO

Información del Paciente

Datos Básicos:
Nombres: Luz Angela
Apellidos: Quintero Pinos
Número de documento: 31189453

Edad: 33 Sexo: ☒ F ☐ M

Información Traslado
Origen: Universidad Icesi
Destino: Clínica de Occidente

Información Clínica
Alergias: Ninguna
Nivel de urgencia: ☐ Vital ☒ No Vital
Procedimientos Realizados:
Resumen de urgencia:
Cuadro de diabetes, niveles altos de azúcar.

Signos Vitales

Pulso: 80 /minuto

Frecuencia Respiratoria: 18 /min

Presión Arterial: 130/90 mm/Hg

Temperatura: 37 °C

Información Traslado

Ver Traslados

Atendiendo Paciente

Escriba el mensaje aquí

Enviar Mensaje

FIGURA 51. PANTALLA DATOS DE PACIENTE EN CAMINO

Confirmar Ingreso	Paciente	Estado	Resumen de Urgencia	Tipo de Urgencia	Responsable	Acompañante	Traslado
<input checked="" type="checkbox"/>	Luz Angela Sarria	En Camino	3	SOAT	Comfenalco	Amana Quintero	Ver Paciente
<input checked="" type="checkbox"/>	Luz Angela Sarria	En Camino	3	SOAT	Comfenalco	Amana Quintero	Ver Paciente
<input checked="" type="checkbox"/>	Luz Angela Sarria	En Camino	3	SOAT	Comfenalco	Amana Quintero	Ver Paciente
<input type="checkbox"/>	Luz Angela Sarria	En Camino	3	SOAT	Comfenalco	Amana Quintero	Ver Paciente
<input type="checkbox"/>	Luz Angela Sarria	En Camino	3	SOAT	Comfenalco	Amana Quintero	Ver Paciente
<input type="checkbox"/>	Luz Angela Sarria	En Camino	3	SOAT	Comfenalco	Amana Quintero	Ver Paciente
<input type="checkbox"/>	Luz Angela Sarria	En Camino	3	SOAT	Comfenalco	Amana Quintero	Ver Paciente

FIGURA 52. PANTALLA DE LISTA DE SERVICIOS ATENDIDOS

7.11. COMPONENTE DE BACK END

El componente de Back-End es el encargado de realizar las tareas que demandan los agentes del servicio utilizando una base de datos para guardar la información intercambiada en cada transacción de los usuarios y una serie de servicios web que permiten la comunicación entre los involucrados y el Centro Automático de Despacho. Este componente consiste en 3 partes importantes, una infraestructura robusta que permita atender solicitudes de manera concurrente y estable, con la cual tener tiempos de respuesta apropiados para el contexto manejado, una base de datos, basada en un modelo relacional que permita almacenar y gestionar la información generada por los agentes de la mejor manera posible para ofrecer un servicio de calidad que cumpla con los requerimientos planteados; por último se tiene el CAD,

el cual se basa en la utilización de una serie de servicios web que permiten atender las solicitudes de los usuarios y poder realizar cada una de las tareas planteadas, posibilitando el cumplimiento de los requerimientos funcionales. En el diagrama 52, se pueden apreciar los componentes de la arquitectura del Back-End:

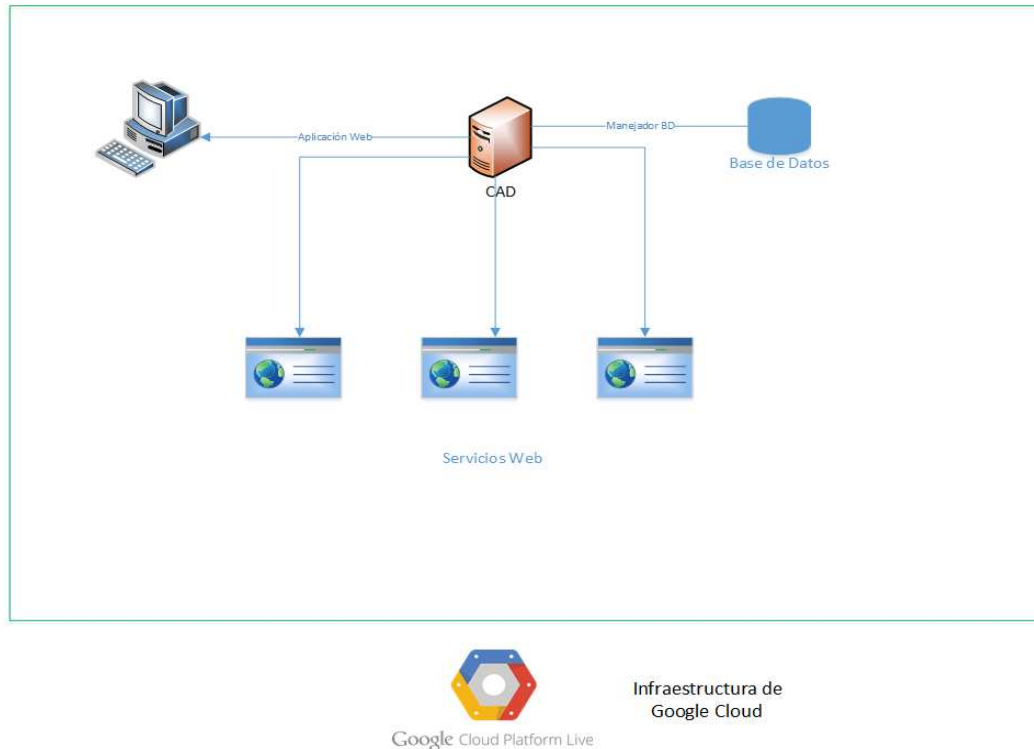


FIGURA 53. COMPONENTES DE ARQUITECTURA.

7.11.1 SOFTWARE DE DESARROLLADO INSTALADO

Para el desarrollo de la funcionalidad se utilizaron diferentes herramientas de software los cuales son:

- **My SQL WorkBench:** Es una plataforma visual que unifica el diseño y la implementación de arquitecturas de base de datos, provee soluciones para realizar modelos de datos, desarrollo SQL, herramientas de administración para configuración de servidores, administración de usuarios, back up, entre otros. Este programa fue elegido para realizar pruebas de base de datos de manera local (Oracle, 2015). Es un programa disponible para Windows, Linux y MAC OS X y se contiene 5 herramientas principales para desarrolladores:
 - Diseño: Permite diseñar una base de datos relacional utilizando una herramienta de diseño visual con la cual podemos modelar, generar y gestionar nuestras bases de datos.
 - Implementación: Provee una serie de herramientas para crear, ejecutar y optimizar consultas SQL.

- Administración: Provee una consola visual para administrar fácilmente los ambientes MySQL y tener una mejor visibilidad de mis bases de datos y poder gestionarlas.
 - Visual Performance Dashboard: Todo un conjunto de herramientas que ayudan a mejorar el rendimiento de las aplicaciones basadas en MySQL.
 - Migración de Bases de Datos: Ahora provee un conjunto de herramientas para hacer migraciones de datos de SQL Server, Microsoft Access, Sybase ASE, a una base de datos MySQL.
- **Eclipse:** Eclipse es famoso por ser un entorno de desarrollo integrado Java (IDE), especializado para el lenguaje Java, pero también se puede utilizar con lenguajes C, C++ y PHP. Uno puede combinar fácilmente soporte de idiomas y las diferentes características que ofrece eclipse en cualquiera de los paquetes por defecto. Eclipse facilita el proceso de desarrollo al permitir crear proyectos de java y clases de una manera sencilla y rápida, además, de permitir una integración con el SDK (Software Development Kit) de Google para poder implementar aplicaciones que serán subidas a la plataforma de Google Cloud. Una última ventaja del Eclipse es la posibilidad de integrar el desarrollo de aplicaciones que utilizan librerías de Java y de Google en un solo proyecto, integrando ambos SDKs. De ser necesaria una nueva librería, esta se puede importar sin mayores inconvenientes al proyecto utilizando las ayudas visuales del ambiente de desarrollo eclipse.
 - **Google Cloud App Engine:** Google App Engine es una plataforma como servicio (PaaS, por sus siglas en inglés) que permite crear y ejecutar aplicaciones en la infraestructura de Google. Las aplicaciones ejecutadas por el App Engine son fáciles de construir, fáciles de mantener y fáciles de escalar a medida que cambian las necesidades de tráfico y almacenamiento de datos. Con App Engine, no hay servidores para su mantenimiento, sólo tiene que cargar su aplicación y ya está lista para entrar en funcionamiento. Google App Engine soporta aplicaciones escritas en diversos lenguajes de programación como:
 - Java: Uso de entorno de ejecución de Java de App Engine, usted puede construir su aplicación utilizando tecnologías Java estándar.
 - Python: App Engine cuenta con un intérprete de Python rápido y bibliotecas estándar de Python.
 - PHP: App Engine utiliza los servicios de plataforma en la nube de Google bajo el capó cuando se llama a las funciones estándar de PHP.
 - Go: App Engine ofrece un entorno de ejecución Go que funciona de forma nativa de código compilado Go.

Con la utilización de estos 3 programas, el SDK de Google y el JDK de Java, se desarrolla el componente de BackEnd basado principalmente en la base de datos, la implementación de Web Services y el manejo de los mensajes entre las aplicaciones implementadas para el componente de Front End.

7.11.2 PREPARACIÓN DE AMBIENTE DE DESARROLLO

En esta sección se describe el proceso necesario de preparación de un ambiente de desarrollo que permitiera implementar el servicio SIEF en sus componentes de Back End.

- **My SQL WorkBench:** El proceso de instalación y configuración de este programa tiene una duración de aproximadamente 4 horas, el diagrama de flujo explica el proceso realizado en la instalación. Figura 53

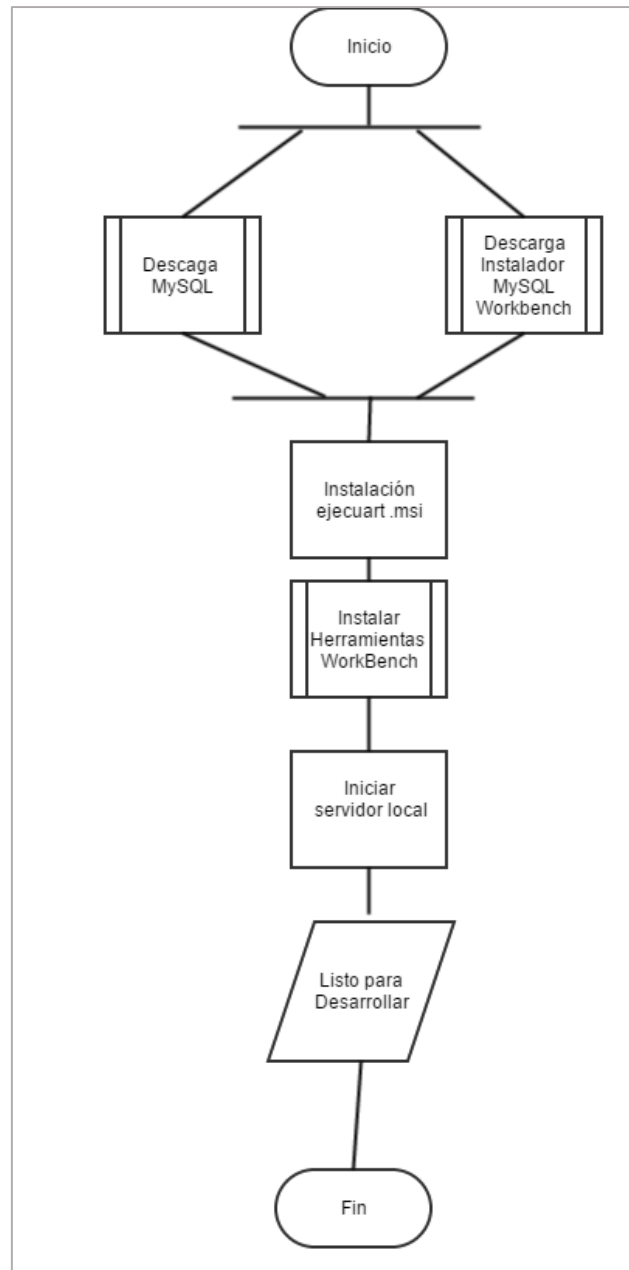


FIGURA 54. INSTALACIÓN MYSQL

Para el desarrollo del Sistema propuesto, se instaló la herramienta WorkBench con los kits para diseño, implementación y gestión.

- **Java Development Kit:** Este es un kit de desarrollo necesario para poder ejecutar cualquier aplicación que funcione con lenguaje Java. Para el desarrollo de la solución se descargó e instaló la versión 1.7.0_75, la cual es la recomendada para programar aplicación web para subir a la plataforma del App Engine de Google. El proceso de instalación es el siguiente, ver figura 54.

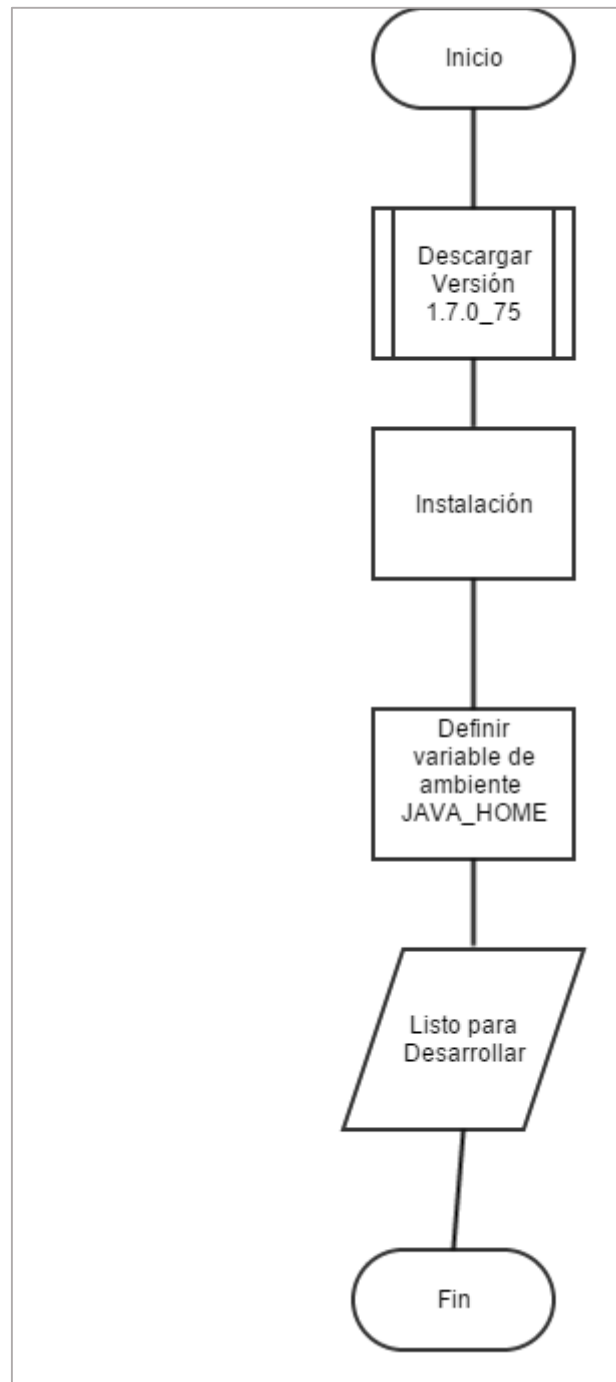


FIGURA 55. INSTALAIÓN JDK

- **Software Development Kit:** Los complementos o kit de desarrollo son necesario para realizar el proyecto, para Instalar el SDJ existen dos caminos los cuales se ilustran en el figura 55.

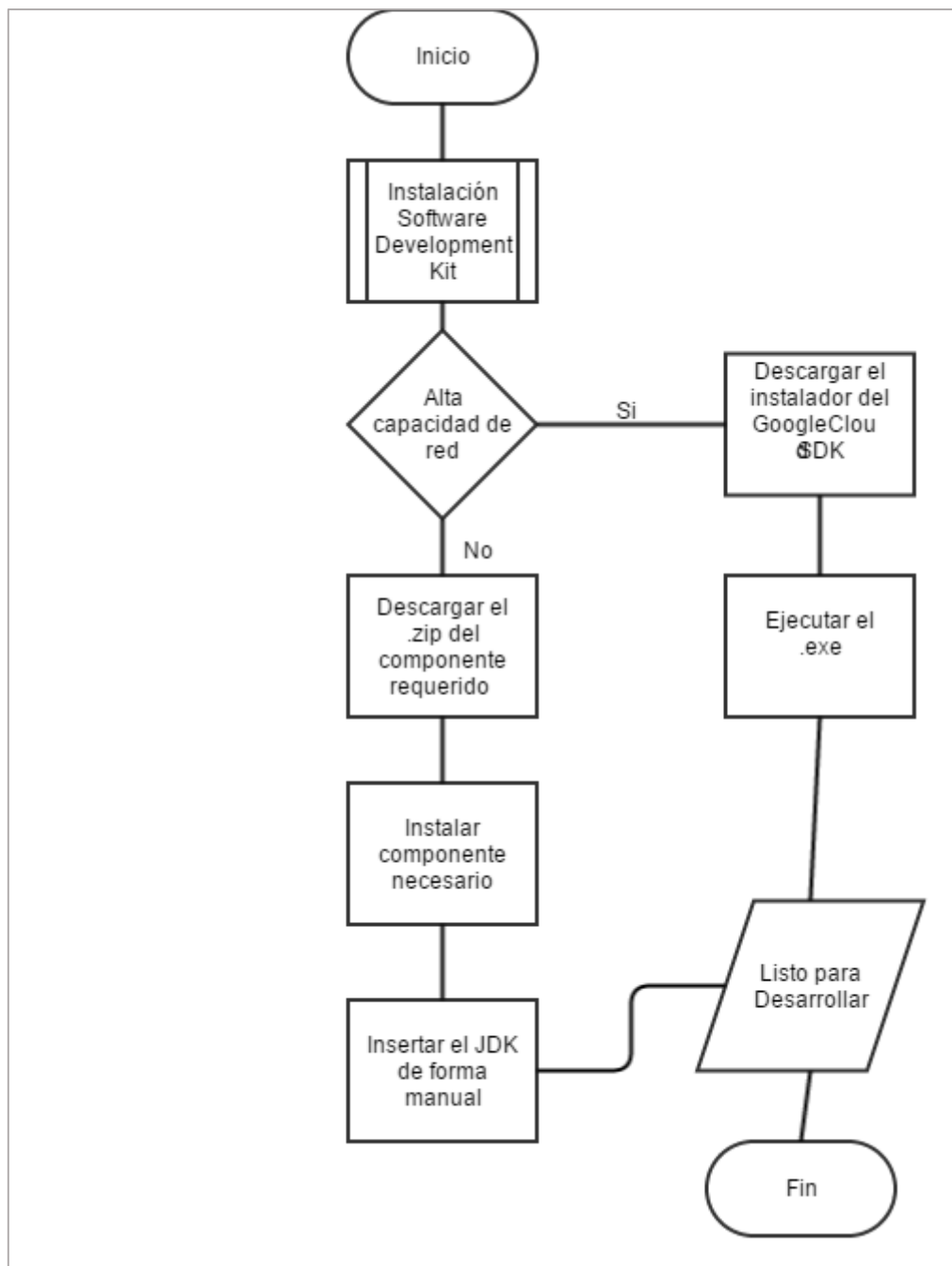


FIGURA 56. INSTALACIÓN DEVELOPMENT FIR.

En este trabajo se decide utilizar el procedimiento de alta capacidad de red, dada la simplicidad de su ejecución. Una vez instalado el SDK, en la consola de administración, en la cual se dejan seleccionadas todas las librerías del AppEngine y las librerías para realizar el Deploy.

- **Eclipse:** El proceso de instalación del eclipse toma alrededor de 6 horas contando con la integración con el JDK de Java y el SDK de Google, además, de la instalación de las librerías necesarias para ejecutar una aplicación Web de Google de manera local y poder subirla a la plataforma de Google Cloud en tan solo un clic.

El diagrama de flujo de la figura 56 explica el proceso de instalación general de Eclipse.

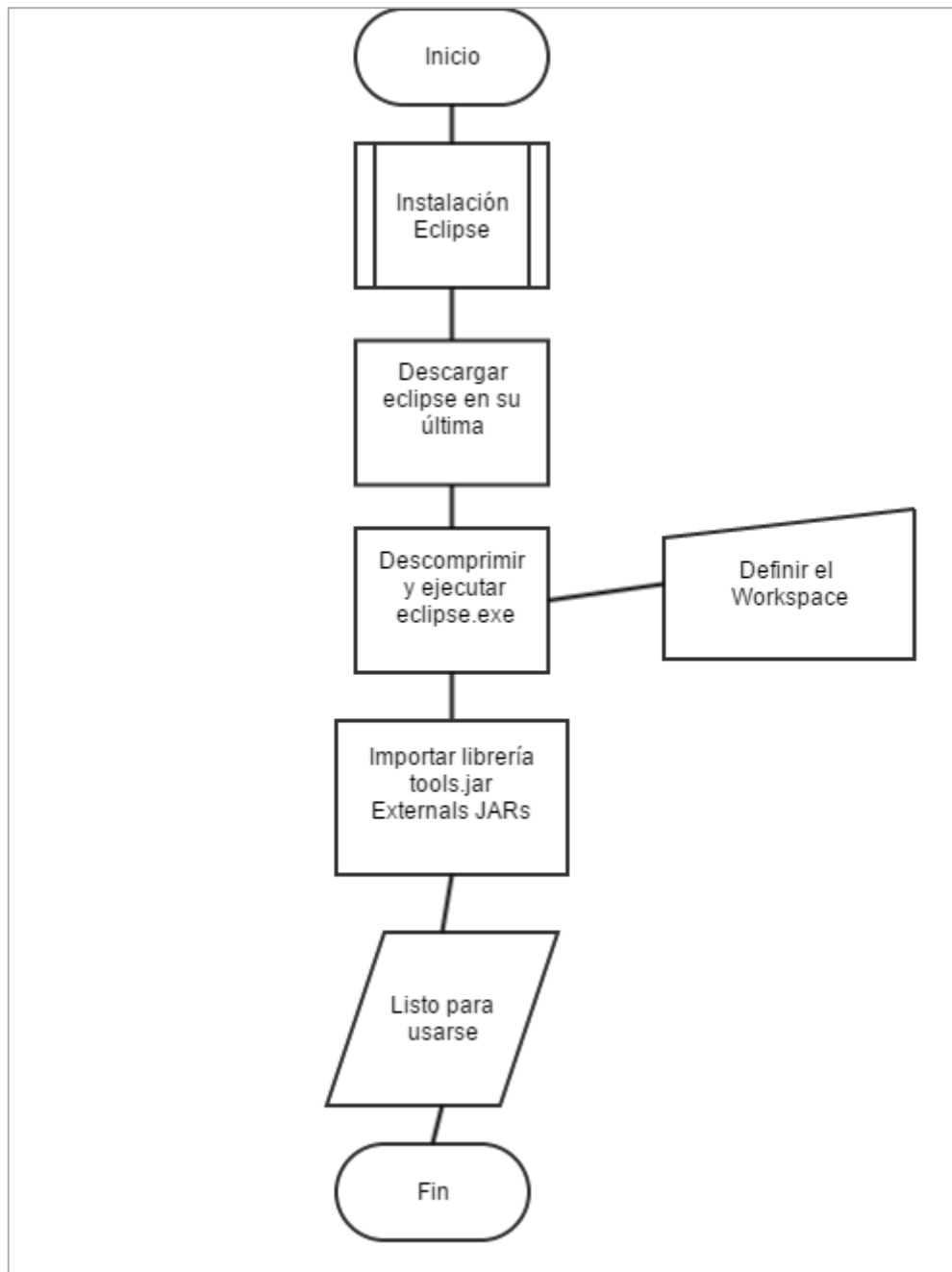


FIGURA 57. INSTALACIÓN ECLIPSE

Una vez tenemos estos datos, es necesario realizar la integración con el SDK de google. Para esto se debe instalar una serie de Plugins en el eclipse, existen plugin diferentes para cada una de las versiones del eclipse, así, se debe revisar primero la versión del eclipse que se está utilizando y dirigiarnos a la sección “Using the Google Plugin for

Eclipse”, donde encontraremos un enlace para descargar el plugin junto con el proceso de instalación. En la siguiente tabla se resume lo mencionado:

Eclipse version	Installation instructions	Direct plugin link
Eclipse 4.4 (Luna)	Plugin for Eclipse 4.4 (Luna)	https://dl.google.com/eclipse/plugin/4.4
Eclipse 4.3 (Kepler)	Plugin for Eclipse 4.3 (Kepler)	https://dl.google.com/eclipse/plugin/4.3
Eclipse 3.8/4.2 (Juno)	Plugin for Eclipse 3.8/4.2 (Juno)	https://dl.google.com/eclipse/plugin/4.2

TABLA 14. VERSIONES ECLIPSE.

Para instalar el plugin se recomienda utilizar el asistente de instalación de software de eclipse, ya que es un proceso muy simple para el usuario. En eclipse nos dirigimos a help>Install >New Software y se pega el enlace del plugin. Una vez pegado el enlace, se seleccionan todos los plugins a instalar y se inicia el proceso. Este proceso toma 3 horas debido a que debe descargar varios plugins, instalarlos y aplicarlos al eclipse. Una vez termina de aplicar todos los plugins el eclipse se reinicia y ya podremos crear proyecto de Google Web Application integrados con el lenguaje Java.

El último paso de configuración de la herramienta eclipse, es modificar el archivo eclipse.ini, en el cual le especificamos al eclipse que ejecute la máquina virtual de Java al iniciar, para que las herramientas de Google puedan realizar el Deploy de la aplicación desarrollado con el proyecto ubicado en el Google App Engine. A continuación se presenta la línea que debe agregarse:

```
-vm  
C:\Program Files\Java\jdk1.7.0_75\bin\javaw.exe
```

- **App Engine:** La configuración del ambiente App Engine se resume en el diagrama que explica el flujo de la aplicación.

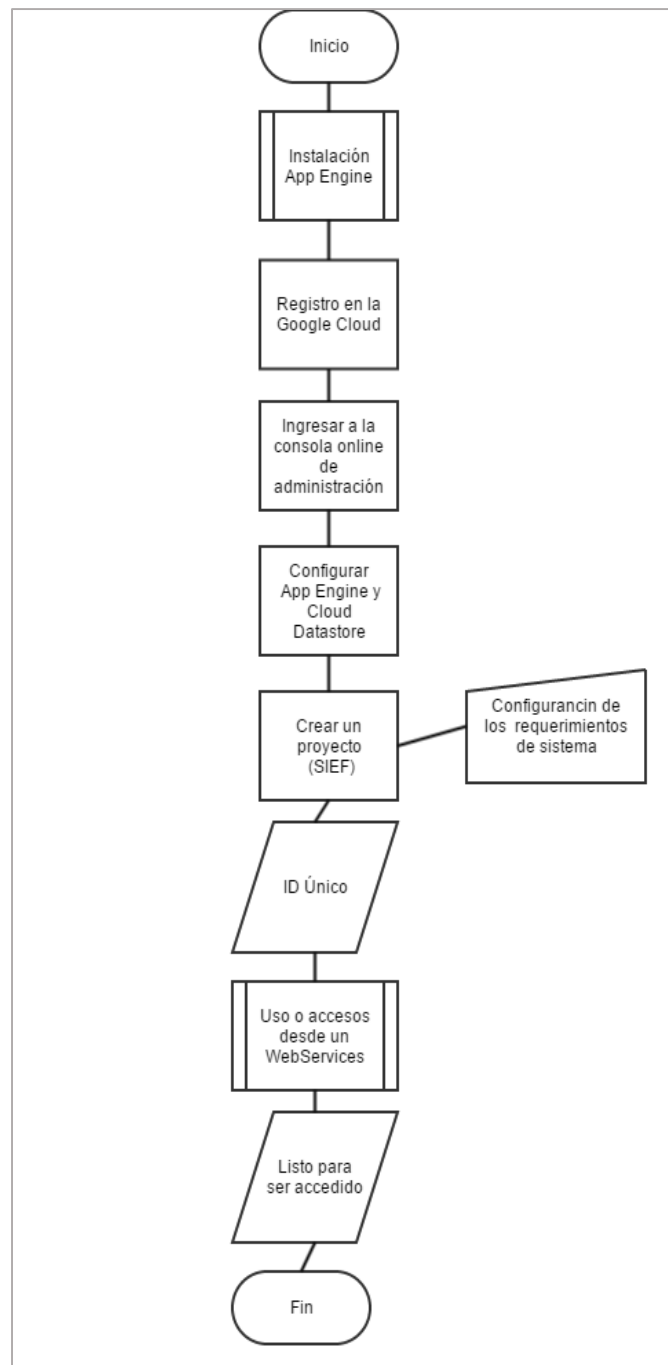


FIGURA 58. INSTALACIÓN APP ENGINE

La configuración de los requerimientos de sistema son: espacio de almacenamiento en disco, configuración de back ups, cantidad aproximada de solicitudes por mes, entre otras. Como el piloto de la propuesta se realiza bajo un ambiente controlado, no son necesarios muchos recursos físicos para ejecutar la aplicación, por ende se configura el proyecto con los requerimientos mínimos.

7.12. IMPLEMENTACIÓN BASE DE DATOS

Para la implementación de la base de datos se utilizó el programa MySQL Workbench, usando la herramienta de modelado de datos, con la cual fue posible generar un script para crear la base de datos. Una vez se tenía este script se crea una base de datos relacional en el espacio de Google Cloud. En esta sección se explicará todo el proceso realizado para tener una base de datos con la cual poder guardar la información de las instituciones involucradas en el sistema, los empleados, ambulancias y toda la información intercambiada entre las partes para lograr tener registros que puedan ser consultados en caso de auditorías. Este desarrollo es crucial para el sistema, ya que establece los cimientos para tener un sistema estable.

MySQL Workbench tiene un conjunto de herramientas que ayudan con el diseño del modelo de datos, permitiendo hacer en la herramienta un modelo relacional de base de datos, como el presentado en la sección de diseño. Debido a que no fue posible importar el diagrama hecho con el programa Visio, toco utilizar la herramienta de modelado y realizar el diagrama dentro del programa MySQL Workbench nuevamente, incluyendo las tablas definidas, los campos, las llaves primarias y foráneas. Este diagrama queda más completo y se aprecia mejor las relaciones entre cada una de las tablas, además de cumplir con los estándares de diseño. En la figura 58 se puede apreciar una parte del modelo generado:

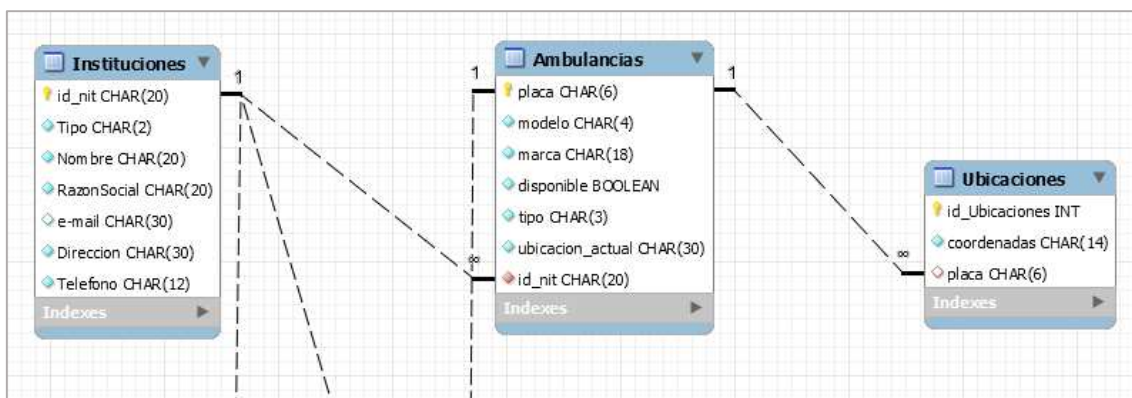


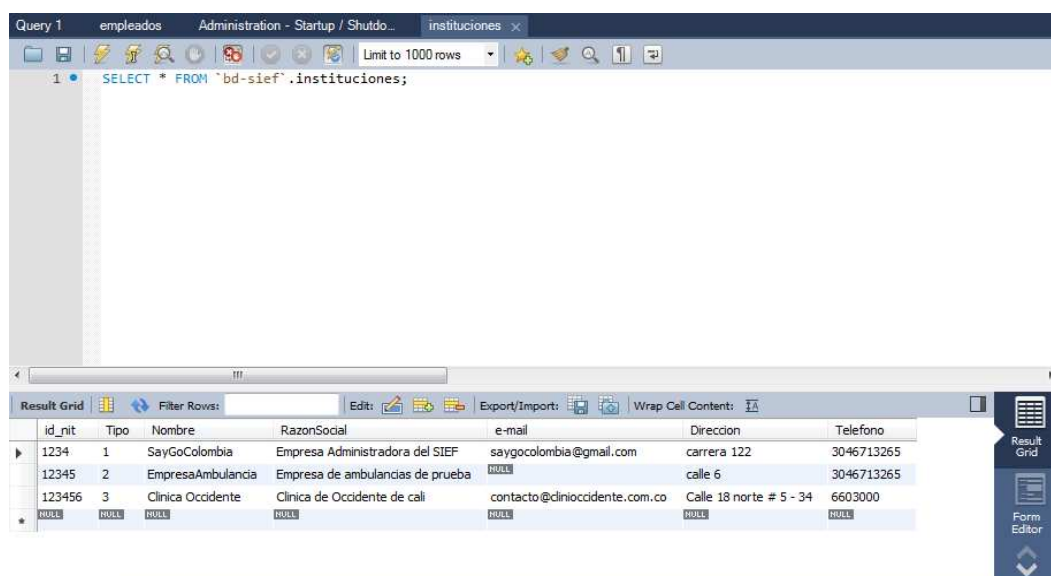
FIGURA 59. MODELO RELACIONAL BASE DATOS

Para ver el Modelo relacional generado, puede dirigirse al anexo Modelo Relacional WorkBench.png, para tener más claridad del proceso.

Una vez realizado el modelo, se verifica si es posible generar una base de datos a partir del diagrama. Este paso resulta ineludible porque es posible encontrar errores de diseño, como por ejemplo, tipo de datos, claves foráneas mal asignadas, entre otros errores que generan problemas en la implementación de la base de datos. Cuando se detectan y corrigen los errores de diseño, utilizando el programa, generamos un script en lenguaje SQL que representa el modelo relacional. Este script nos crea las tablas con sus atributos y relaciones de manera automática en el momento de ser ejecutado, se puede ver el script completo en el anexo “ScriptCreation-BD-SIEF.sql”

Ahora bien, se pasa a utilizar la herramienta de desarrollo del MySQL WorkBench, porque al pasarle el script de creación de la base de datos, esta herramienta leerá el archivo .sql y ejecutará cada una de las sentencias SQL dentro del archivo y automáticamente creará la base de datos con todas las tablas y relaciones definidas, en un servidor local. Este proceso toma alrededor de 20 minutos, y al finalizar el proceso tendremos nuestra base de datos creada.

En este modelo ya se tiene la base de datos del SIEF en un servidor local de pruebas, pero al no tener un desarrollo de conexión con la base de datos, se necesita utilizar la función de gestión de base de datos instalada con el MySQL WorkBench. Con esta herramienta se pueden usar sentencias SQL para consultar los datos de las tablas, añadir nuevos registros, eliminar registros y todas las aplicaciones básicas de sentencias SQL. Pero, la principal razón para utilizar la herramienta administrativa de base de datos, es que permite ejecutar estas funciones desde un ambiente gráfico amigable, además, de ir visualizando como van quedando las tablas y una vez estemos seguros de los cambios, se efectúan las sentencias SQL generadas y los cambios serán aplicados en la base de datos. Esta herramienta de administración nos permite gestionar la base de datos y poder ver los cambios antes de aplicarlos, así, de una manera gráfica y fácil, modificar una base de datos de la mejor manera. Con la herramienta de administración se realizaron una serie de registros en cada una de las tablas para comprobar que los datos se agregaran de manera correcta. A continuación se muestra como ejemplo la tabla institución, y como se ven los registros desde la herramienta:



id_nit	Tipo	Nombre	RazonSocial	e-mail	Direccion	Telefono
1234	1	SayGoColombia	Empresa Administradora del SIEF	saygocolombia@gmail.com	carrera 122	3046713265
12345	2	EmpresaAmbulancia	Empresa de ambulancias de prueba	NULL	calle 6	3046713265
123456	3	Clinica Occidente	Clinica de Occidente de cali	contacto@clinioccidente.com.co	Calle 18 norte # 5 - 34	6603000

TABLA 15. MUESTRA TABLA INSTITUCIÓN

Como se aprecia en la imagen, se tienen dos ventanas principales, una con las sentencias SQL y la otra con el resultado. Adicionalmente, se aprecia la tabla de instituciones con 3 registros, con los cuales se hará la prueba piloto del sistema.

Al culminar los procedimientos anteriores, se tiene una base de datos estable y lista para entrar en funcionamiento, pero se encuentra alojada en un servidor local, en el computador donde está instalado el programa MySQL WorkBench, por ende, es necesario exportar la base de datos utilizando la función de desarrollo. Existen 2 maneras de exportar una base de datos para poder migrar la base de datos al google cloud, la primera es exportar un archivo de proyecto tipo Dump

y la segunda es exportar un archivo .sql con todas las sentencias necesarias para generar la base de datos del SIEF en otro lugar. Para migrar la base de datos, se utilizó la segunda forma de exportación de base de datos, debido a que a la hora de importar la base de datos en el Google Cloud, este solo permite archivos .sql, para ejecutar las sentencias y crear una base de datos.

Ahora bien, se tiene un archivo .sql, el cual debe ser importado al servidor ubicado en Google Cloud, es decir, al proyecto creado previamente. Este proceso se inicia seleccionando el proyecto al que queremos importar la base, una vez ahí, debemos habilitar la función de Cloud Storage, la cual funciona como la aplicación Google Drive (Aplicación orientada a brindar almacenamiento en la nube) de usuarios finales, pero orientado a servidores y aplicaciones web. En el espacio del Cloud Storage se pueden guardar todos los archivos que utilizan las aplicaciones tales como imágenes, documentos de texto, entre otros; en nuestro caso subimos el archivo .sql, para que el Datastore de Google pueda encontrar la ruta del archivo. Para subir el archivo .sql, con las sentencias de creación de la base de datos, al Cloud Storage solo es necesario, buscarlo en el directorio local y arrastrarlo hasta la ventana del navegador donde esté la consola de administración del Google Cloud Storage del proyecto donde se desee almacenar dicho archivo; como punto de referencia, el lector puede hacer un símil al proceso de guardar un archivo en la plataforma de Google Drive o Dropbox.

Por último, el archivo que genera la base de datos del SIEF está almacenado en la nube, pero no es posible utilizarlo para la implementación de la base de datos, es necesario generar una instancia en el proyecto. Esta instancia del proyecto especifica las propiedades de la base de datos del proyecto, por lo que toca configurar en la plataforma Google Cloud el lugar donde se quiere generar la base de datos -este lugar hace referencia al país donde está ubicado el servidor físico, los recursos de memoria con los que se creará la instancia, el plan de facturación y las copias de seguridad programadas para que la información registrada tenga un back-up en caso de ocurrir un incidente. En la siguiente figura se logra apreciar la configuración de la instancia usada en la prueba piloto:

Crear instancia de Cloud SQL

ID de la instancia de Cloud SQL [?]
Utiliza solo letras en minúscula, números y guiones.

peak-vortex-87808:

Región [?] Nivel [?]

Estados Unidos D1 (512 MB de RAM)

Opciones

Versión de la base de datos

MySQL 5.5

Plan de facturación D1

☒ Por uso — 0,10 \$ por hora
Se te cobrará por cada hora en que se ejecute la instancia.

☐ Paquete — 1,46 \$ al día
Se efectuarán cargos mensuales por el número de días en que exista la instancia de la base de datos.

Ubicación preferida ID de la aplicación de App Engine

Seguir la aplicación de App Engi... peak-vortex-87808

Copias de seguridad
Todas las horas están en UTC-5

☒ Habilitar las copias de seguridad 14:00 — 18:00

☐ Registro binario (requiere copias de seguridad)

FIGURA 60 CONFIGURACIÓN PRUEBA PILOTO.

Una vez generada una instancia para el proyecto, es posible ejecutar las sentencias SQL generadas en el archivo de creación de la base de datos. Adicionalmente, con la creación de ésta será posible ejecutar las consultas y los métodos que se usarán posteriormente en el proyecto de App Engine.

En este punto del desarrollo ya se cuenta con una base de datos MySQL lista para funcionar en la plataforma de Google Cloud, pero existe un último elemento clave que es importante mencionar el cual es: la caja de arena o sandbox. La caja de arena es un concepto muy importante a la hora de implementar aplicaciones con el App Engine debido a que los proyectos creados bajo el ambiente del App Engine se ejecutan en una máquina virtual en servidores de google, junto con miles de otras aplicaciones, algunas que consumen pocos recursos como la del servicio SIEF y otras más demandantes como la aplicación Evernote, así, resulta crucial definir la instancia y la caja de arena para el proyecto y garantizar que los objetos, hilos y transacciones generadas por el sistema sean ejecutadas de la mejor manera en nuestro ambiente de desarrollo sin afectar las demás aplicaciones vecinas. Cabe aclarar, que google no permite que la información de una aplicación se mezcle con otra, bloqueando y descartando cualquier información que no pertenezca al hilo de ejecución de la aplicación, así, es de vital importancia definir las instancias y la caja de arena para que los objetos y transacciones del SIEF siempre sean instanciados cumpliendo las normativas de Google Cloud, para garantizar que ningún dato se pierda durante la ejecución de métodos y tareas.

Con estos procedimientos ya se tiene una base de datos lista para agregar, buscar, modificar y eliminar información al sistema. En resumen se utilizó el programa MySQL WorkBench para facilitar el proceso de desarrollo de la base de datos, usando herramientas gráficas que permiten que personas con conocimientos básicos de base de datos puedan generar una base de datos robusta, escalable y funcional, gracias al asesoramiento a la hora de diseñar, administrar y desarrollar nuestra base de datos. Una vez se tienen los archivos .sql, se debe entender el funcionamiento de la plataforma de Google Cloud y el App Engine para garantizar un funcionamiento óptimo del servicio.

7.13. IMPLEMENTACIÓN DE SERVICIOS WEB.

Para facilitar el procesamiento de los dispositivos móviles, se consideró el desarrollo de Web Services para realizar las tareas que demandan procesamiento ya que estos dispositivos cuentan con recursos de hardware limitados que deben ser compartidos con un conjunto de aplicaciones, por ende, se toma la decisión de minimizar el procesamiento local para mejorar el rendimiento y producir una mejor experiencia como usuario al momento de cargar e intercambiar la información. Otra ventaja es la posibilidad de tener un procesamiento centralizado usando la infraestructura de Google y la plataforma del App Engine, la cual ajusta dinámicamente los recursos según la demanda, es decir, entre más solicitudes, hilos y transacciones estén generando los usuarios al CAD, Google aumentará los recursos de la máquina virtual donde se esté ejecutando la aplicación para que pueda responder a las solicitudes sin afectar lo estipulado en los requerimientos funcionales, y así, garantizar una aplicación estable y robusta. En esta sección del documento se explicará el proceso de desarrollo de los Servicios Web para el Storage, Usuario, Paramédicos e instituciones médicas. Ver figura 60.

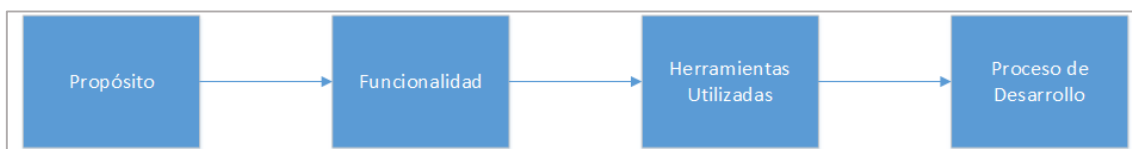


FIGURA 61. MANEJO DE PROCESOS

7.13.1 SERVICIO WEB ALMACENAMIENTO

Este Servicio Web está orientado a manejar la información que será registrada y modifica en nuestra base de datos ubicada en el Google App Engine. Fue desarrollado en 5 etapas: la primera se centra en el manejador con el Datastore, la segunda en la declaración de las clases, la tercera en la generación del servicio web y el archivo .wsdl, el cuarto es el manejo de los adaptadores de respuesta y la última se centra en el manejo de las solicitudes tipo SOAP.

El propósito de este servicio web es garantizar que las solicitudes de registros y cambios a la base de datos puedan ser realizados de manera ordenada y sin afectar la información existente, realizando una serie de verificaciones, gracias a las cuales es posible garantizar la continuidad de los datos almacenados en la base de datos del sistema SIEF y permitir que se puedan registrar nuevas instituciones, empleados, ambulancias, traslados, paciente, entre otra información que se intercambia en el sistema propuesto. Este servicio web es el que debe implementarse inicialmente, debido a que es el encargado del manejo de los datos que se generan en el sistema.

Las funciones que tiene este servicio se basan en la creación, búsqueda y modificación de los registros enviados a la base de datos. Este servicio web tiene como utilidad guardar la información, generada por un agente, en la base de datos del sistema SIEF, dicha información se almacena en la tabla correspondiente a los tipos de datos enviados y se verifica que cumpla con las relaciones previamente estipuladas, de esta manera se garantiza la buena funcionalidad de la base de datos. Adicionalmente, con este servicio web los agentes pueden buscar la información que necesitan para cumplir con sus tareas, como: buscar información del traslado, el paciente, instituciones cercanas, entre otros, así, este servicio ofrece a los usuarios la posibilidad de realizar tareas de búsqueda de información contenida en el sistema, de manera remota con solo enviar una petición al CAD. La última funcionalidad crucial de este servicio es la posibilidad de modificar información ya registrada en la base de datos, aunque la mayoría de los datos ingresados al sistema no pueden ser modificados, existe información dentro del sistema que puede ser modificada como: el estado del traslado, las enfermedades diagnosticadas al paciente, los procedimientos realizados, los estados del paciente y la ubicación actual de la ambulancia; gracias a este servicio web se da garantía que los agentes sólo puedan modificar la información que está permitida para ser cambiada, de lo contrario será bloqueada su solicitud.

Como se menciona en la introducción de esta sección del documento, el desarrollo de este servicio web se elabora en 5 etapas principales, cada una de ellas será descrita para dar claridad sobre el proceso de implementación del servicio ofrecido. A continuación se muestra una figura que ilustra el proceso general:

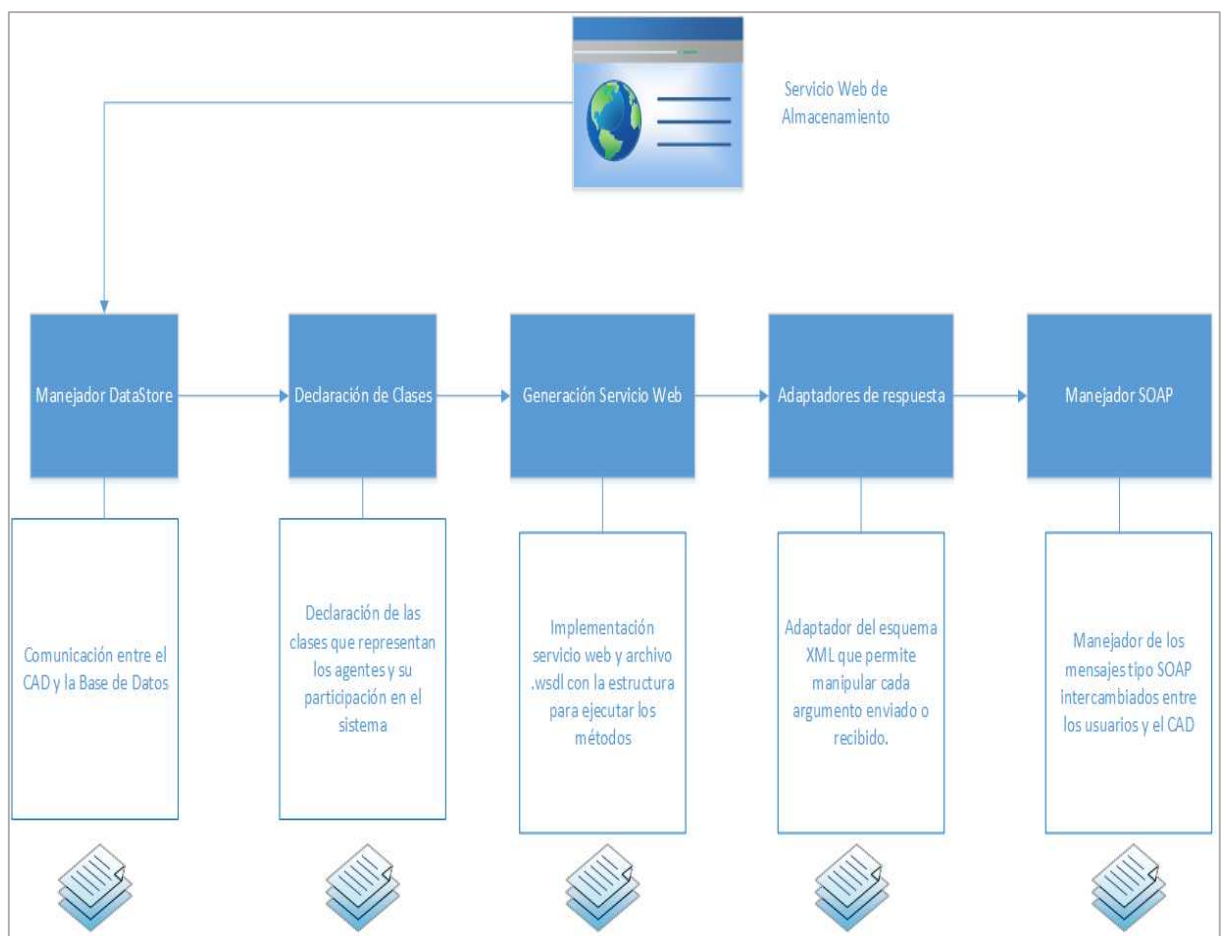


FIGURA 62. MODELO GENERAL ALMACENAMIENTO EN SERVICIO WEB

En la primera etapa de desarrollo del servicio web nos centraremos en realizar las implementaciones necesarias para lograr que un conjunto de datos pueda ser registrado en nuestra base de datos a través del DataStore de Google Cloud. Esta fase es necesario implementarla porque el manejador de la base de datos es la única entidad del servicio que debe tener acceso a los datos contenidos en la base de datos, y el resto de aplicaciones debe utilizar los métodos ofrecidos por esta clase para registrar, buscar o modificar un registro en la base de datos. Este elemento es una primera etapa para la implementación de un servicio web seguro que garantice la continuidad de la información almacenada.

Inicialmente se debe manejar la librería del DataStore, la cual se encuentra importada después de utilizar el plugin del SDK de Google para eclipse. Este conjunto de clases se encarga de establecer la comunicación con la base de datos, previamente instanciada en el proyecto SIEF del APP Engine, y de actualizar los registros que generan las aplicaciones sobre esta, además, permite realizar búsquedas y eliminar una entrada. Se inicializa una variable estática de tipo DataStore, la cual representará la conexión con la base de datos. Esta clase necesita manejar llaves únicas para cada una de las entidades (son la representación de una entrada o registro en la base de datos) que serán gestionadas por el DataStore, dichas llaves se basan en un tipo -es el nombre de la tabla, y el campo con la llave primaria, es decir el id único de la entidad. De esta manera, si se desea crear un institución se debe pasar los datos completos de la institución, generar una nueva entidad en el CAD con la tabla destino y la llave primaria, y luego utilizar el comando `updateEntity()` del Datastore, así, se carga la entidad a la base de datos y la información de dicha entidad es almacenada. En la siguiente tabla se aprecia cómo queda guardada la información en la base de datos del SIEF:

Consultar por tipo ▼

Tipo

Institucion ▼

Filtros

Entidades de Institucion

Eliminar

<input type="checkbox"/> Nombre/ID	email	id_nit	latitud	longitud	nombre	razonSocial	telefono
<input type="checkbox"/> name=1151949192	c0ccci@clini0.com	1151949192	-77.26763185000004	1.1989962003323649	CliniOccidente	1151949192	4442233
<input type="checkbox"/> name=1234	saygocolombia@gmail.com	1234	10	10	SayGoColombia	Empresa Administradora del SIEF	+573046713265
<input type="checkbox"/> name=12345	empresaAmbula@gmail.com	12345	27	27	Empresa de Ambulancias	Empresa de Ambulancias de Prueba	3046713265

TABLA 16. TABLA BASE DE DATOS SIEF

El siguiente diagrama ayuda a comprender mejor el proceso realizado:

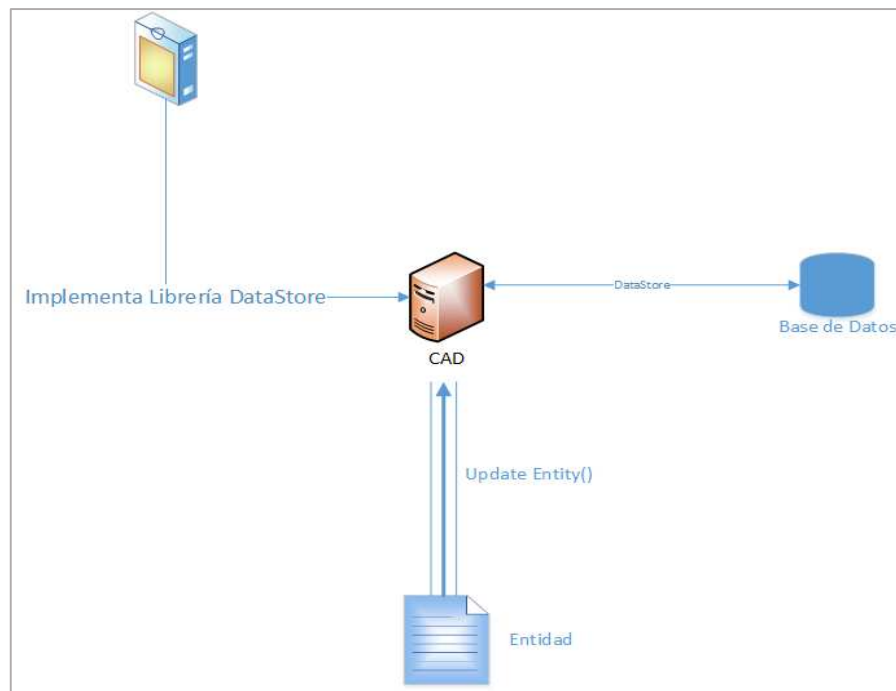


FIGURA 63. ACLARACIÓN DE PROCESOS

Otro proceso importante a tener en cuenta durante el desarrollo de esta etapa, son las consultas de búsqueda. Gracias al DataStore de google no es necesario la implementación de complicadas sentencias SQL para buscar los datos necesarios a la hora de atender la solicitud de un usuario, se debe buscar la entidad utilizando la llave formada por el tipo y el dato primario. El DataStore busca en la tabla indicada la llave enviada, y cuando la encuentra retorna la entidad solicitada, en caso de no encontrar nada retorna una excepción manifestando que la entidad buscada no existe en la base de datos. Dicha entidad es iniciada localmente mientras se esté utilizando para dar una respuesta rápida y eficiente a las peticiones de los agentes involucrados en el hilo de procesamiento que se está ejecutando. Una vez se tiene la entidad, sólo es necesario pedir las propiedades -los campos- que se necesiten para atender la solicitud, y así, no se utilizan sentencias SQL complejas, que pueden dar cabida a errores, debido a la falta de experiencia en la declaración e implementación de este tipo de sentencias. Al utilizar la librería del DataStore la información se adquiere a partir de una llave y métodos `get()`. Un ejemplo que ayuda a comprender mejor el potencial del DataStore, es la búsqueda de una institución con nit 12345. Se genera una solicitud de tipo Institución y con llave primaria 12345, se utiliza el DataStore para buscar dicha entidad, siguiendo el proceso mencionado previamente. Cuando se encuentra la entidad, procedemos a obtener cada una de las propiedades de la Institución para ser asignadas a un objeto de tipo Institución y finalizar la búsqueda de la institución. A continuación se muestra el código Java necesario para realizar este ejemplo. Figura 63.

```

public static Institucion buscarInstitucion(Key id_nit){
    Institucion institucion = new Institucion();

    try {
        Entity e = ds.get(id_nit);

        institucion.setId_nit(e.getProperty("id_nit").toString());
        institucion.setNombre(e.getProperty("nombre").toString());
        institucion.setRazonSocial(e.getProperty("razonSocial").toString());
        institucion.setEmail(e.getProperty("email").toString());
        institucion.setLongitud(e.getProperty("longitud").toString());
        institucion.setLatitud(e.getProperty("latitud").toString());
        institucion.setTelefono(e.getProperty("telefono").toString());
        institucion.setTipo(e.getProperty("tipo").toString());

        return institucion;
    } catch (EntityNotFoundException e1) {
        System.out.println("No se encontró la Institución con ese Nit"+id_nit.toString());
        e1.printStackTrace();
    }

    return null;
}

```

FIGURA 64. EJEMPLO CÓDIGO BÚSQUEDA

Como se puede apreciar en ningún momento se ejecuta una sentencia SQL de búsqueda, debido a que el DataStore las ejecuta por debajo al utilizar los métodos `get()` y `getProperty()`. De esta manera, fue posible tener una solución confiable en la búsqueda de los datos, sin peligro a incurrir en errores de sentencias SQL, que pudieran afectar el correcto funcionamiento de la base de datos. En resumen, este manejador de base de datos se centra en la creación y búsqueda de registros, ya que SIEF fue un sistema concebido para tener un historial de información robusto que pudiera ser accedido desde cualquier dispositivo y a cualquier hora y, con la ayuda de las librerías de Google fue posible desarrollar e implementar un manejador de base de datos fácil de programar y de gestionar por medio del código java.

La segunda fase se desarrolló utilizando los diagramas de clases de las representaciones de los agentes que interactúan con el CAD, estos agentes son las instituciones, empleados, ambulancias, traslados, pacientes, signos vitales, enfermedades, procedimientos y EPS y los demás agentes involucrados y con participación en el sistema. Con una representación en objetos de lenguaje Java se puede contar con una aplicación estable y funcional que permita manejar la información intercambiada entre las partes, manejar diferentes hilos de procesamiento en simultáneo ofreciendo una aplicación concurrente, para lograr esto se utilizan objetos, que representan al paciente, la ambulancia, el médico y demás objetos necesarios para la representación de los agentes del SIEF. Con esta implementación es posible manejar la información de los traslados, pacientes (signos vitales, enfermedades y procedimientos), solicitudes y navegación de una manera eficaz y rápida. El desarrollo tomó un tiempo aproximado de 1/3 semana.

Con la fase 1 y 2 terminadas, tenemos un sistema capaz de almacenar, buscar y modificar información en la base de datos y un sistema con las clases que representan los agentes involucrados en el servicio ofrecido, pero no se tiene un componente que sepa cómo atender cada solicitud recibida. La tercera fase de este desarrollo se basa en el componente dedicado a ofrecer un método que pueda brindar una solución a cada solicitud de los usuarios, referente al almacenamiento, es decir, el componente del servicio web encargado de entender que el cliente desea guardar un dato, buscarlo o modificarlo; dicho elemento debe luego pasar la información al manejador para cumplir con el requerimiento del cliente. Este es el núcleo del desarrollo de

un servicio web ya que los métodos descritos en esta clase, serán los llamados por los dispositivos móviles y los que realicen la secuencia de pasos necesarios para dar respuesta a la solicitud de los usuarios. El proceso de desarrollo de esta etapa se dividió en 3 partes importantes: los métodos de almacenamiento, métodos de búsqueda, métodos de modificación. En el siguiente diagrama se puede apreciar mejor la funcionalidad de esta tercera fase y el propósito de cada uno de los grupos de métodos definidos:

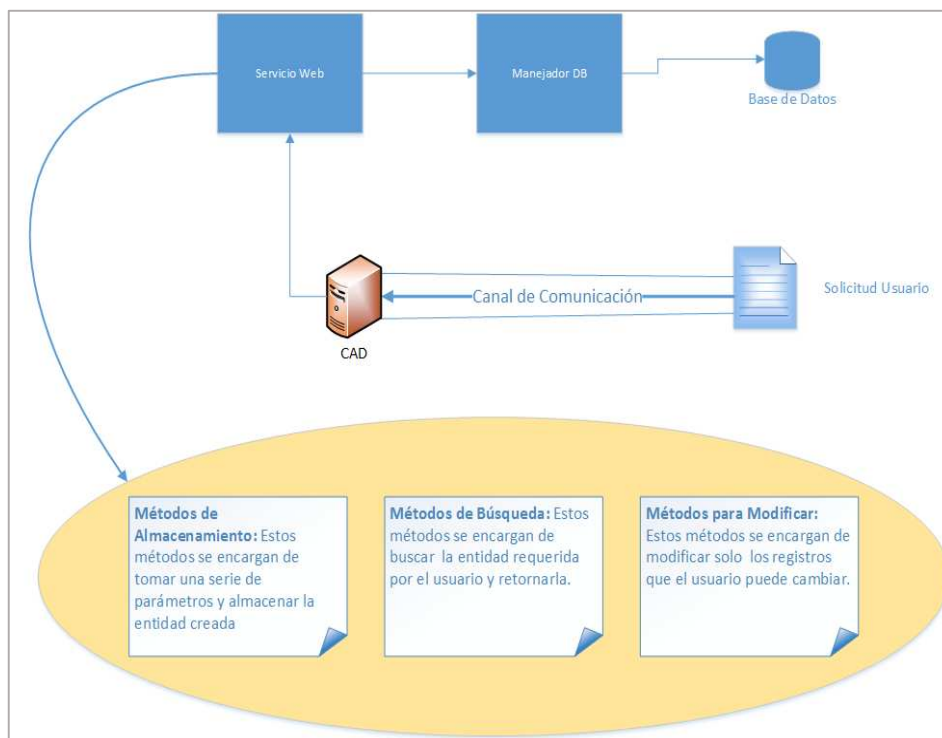


FIGURA 65. MÉTODOS DE ALMACENAMIENTO

El funcionamiento de esta clase en general es tomar una serie de parámetros según el objetivo del método y comunicarse con el manejador de base de datos para que este, escriba o busque la información en la base de datos, así, este servicio web actúa como un intermediario entre el usuario y la base de datos para responder las solicitudes recibidas, y al ser un servicio web, facilita la comunicación con las aplicaciones móviles.

Para que los métodos definidos puedan ser ejecutados correctamente, es necesario definir una estructura del servicio web, esta estructura es definida en un archivo `.wsdl`, el cual describe la estructura de cada uno de los métodos, los parámetros que debe tener y que acciones tomar a la hora de ejecutar cada uno de los métodos. Este archivo funciona como un molde para el servicio web, con el cual asegurarnos de que las solicitudes siempre puedan ser atendidas por algún método definido. Para generar este archivo se utiliza la función `wsgen` (WebServices Generator) del JDK de java, esta utilidad genera el archivo `.wsdl` con la estructura del servicio web definido y el esquema que utilizará para atender las solicitudes entrantes al servidor. Este esquema se genera a partir, de la clase que define los 3 grupos de métodos explicados previamente.

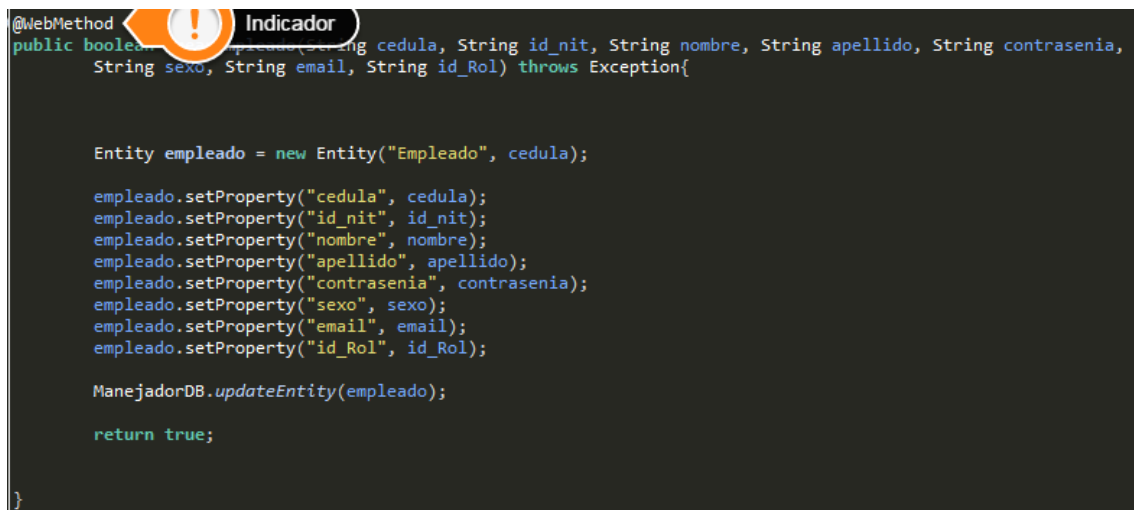
Para poder utilizar esta función se debe especificar la clase de nuestro proyecto que tomará las veces de servicio web, para establecer esta funcionalidad se debe escribir un indicador sobre la

declaración de la clase. Con el indicador `@WebServices`, el programa `wsgen` sabe qué clases deben tener generados una estructura `.wsdl` a partir de los atributos y métodos declarados en esta clase. En la siguiente figura se muestra como debe declararse la clase y el indicador:

```
@WebService
public class WebServiceStorage {
```

FIGURA 66. INDICADOR WEBSERVICE

Adicionalmente, se deben especificar qué métodos serán utilizados en la generación de este servicio web, es decir, qué métodos tendrán una acción en el servicio definido y cuáles podrán ser demandados por los agentes involucrados en el sistema SIEF; para definir dichos método se utiliza el indicador `@WebMethod` sobre la declaración de los mismos. La siguiente figura le puede ayudar al lector a comprender mejor este paso de la implementación del servicio web:



```
@WebMethod
public boolean actualizar(String cedula, String id_nit, String nombre, String apellido, String contrasenia,
    String sexo, String email, String id_Rol) throws Exception{

    Entity empleado = new Entity("Empleado", cedula);

    empleado.setProperty("cedula", cedula);
    empleado.setProperty("id_nit", id_nit);
    empleado.setProperty("nombre", nombre);
    empleado.setProperty("apellido", apellido);
    empleado.setProperty("contrasenia", contrasenia);
    empleado.setProperty("sexo", sexo);
    empleado.setProperty("email", email);
    empleado.setProperty("id_Rol", id_Rol);

    ManejadorDB.updateEntity(empleado);

    return true;
}
```

FIGURA 67. INGRESO MANUAL DE DE USUARIOS AL WEBSERVICE

De esta manera se genera un esquema para el manejo de los servicios web en el cual se establece el namespace (nombre de espacio) el cual define la dirección dentro del proyecto donde se encuentra alojado la clase definida como `WebService`, es importante definir bien esta variable ya que es la ruta que utilizan los paquetes de los clientes a la hora de demandar un método del servidor. Adicionalmente, se genera en la estructura del servicio web los métodos a utilizar y las acciones que se debe el servidor según el nombre del método, y brindarle una respuesta apropiada a las demandas de los usuarios.

Esta fase de implementación tiene una última fase previo a declararse completada, solo resta definir dónde está localizado nuestro servidor y toda la estructura del servicio web definida en las etapas de desarrollo descritas previamente. Para terminar esta última parte de desarrollo se debe configurar la sección del SOAP address, para que los paquetes de las solicitudes sean redirigidos al servicio que debe atender dicha solicitud.

La estructura del SOAP address es la siguiente:

`<soap:address`

`Location="http://Id-Proyecto-App-Engine/NombreClaseServicioWeb+Service`

"/>

En el caso del servicio de implementación, la variable "location" del soap address será la siguiente:

"location=http://peak-vortex-87808.appspot.com/WebservicesDemoService"

Se presenta la dirección de un servicio Demo para evitar problemas en el funcionamiento de la prueba piloto definida para este proyecto. Este servicio Web Demo cuenta con la misma estructura descrita del servicio web de almacenamiento y su desarrollo siguió los mismos procedimientos descritos en las diferentes fases de implementación del sistema SIEF. Si el lector desea estudiar la estructura completa del archivo .wsdl generado, puede dirigirse al enlace presentado a continuación: <http://peak-vortex-87808.appspot.com/WebservicesDemoService.wsdl>

Esta tercera Fase tomó un tiempo aproximado de 3 semanas.

La cuarta fase en la generación de un servicio web funcional, es la generación de un adaptador para el servicio web cuyo objetivo es tomar los argumentos de una solicitud tipo SOAP, obtener su contenido y compartir los parámetros obtenidos con el método del servicio web demandado por el usuario. Una vez comparte estos datos, la clase adaptador debe esperar la respuesta de la clase del servicio web, genera nuevamente la estructura del mensaje SOAP y la respuesta es enviada al usuario. Esta serie de procesos se aprecia de una manera más clara en la siguiente figura:

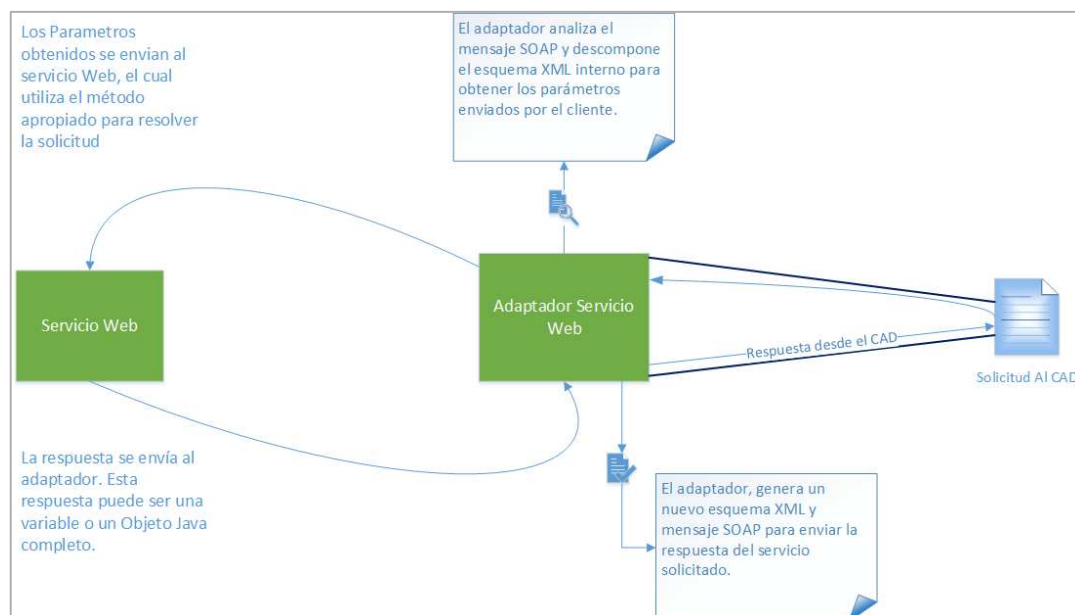


FIGURA 68. EXPLICACIÓN ADAPTADOR SERVICIO WEB

Debido a que esta clase funciona como un intermediario entre la solicitud del usuario y la clase encargada de solucionar la tarea del método demandado, es necesario añadir una serie de clases extras que puedan comprender un esquema de etiquetas XML, para poder obtener los argumentos de la solicitud y armar un nuevo esquema con los argumentos que se envían como respuesta al usuario. Dicha serie de clases extras debe ser de tipo Javax.xml, para poder manejar

la estructura de un archivo .xml, obtener los argumentos y enviar la respuesta utilizando el mismo esquema. Este proceso es algo tedioso, debido a que se deben generar 2 clases por cada método declarado como `@WebMethod`, es decir, si se tiene el método `crearInstitucion()`, debe generarse la clase `CrearInstitucion` y la clase `CrearInstitucionResponse`, cada una con un propósito diferente. La primera, tendrá como objetivo atender la solicitud entrante al CAD, obtener cada uno de los parámetros necesarios para la ejecución del método demandado y enviarlos a la clase encargada de manejar el servicio web como tal; la segunda, se encarga de tomar la respuesta del servidor y crear un nuevo esquema XML con la respuesta que será enviada al usuario. La decisión de implementar esta fase del desarrollo así se toma para garantizar un performance que cumpla con los tiempos de respuesta establecidos, ya que el adaptador no debe buscar en una lista de métodos extensa la estructura necesaria para resolver el archivo XML entrante y puede armar la estructura de la respuesta en un tiempo menor.

Un trabajo arduo en esta fase, permite ofrecer un servicio con mejor experiencia para el usuario y al hablar de traslados en los que la vida de una persona corre peligro, cualquier segundo que se reduzca en las transacciones ayuda, indudablemente, a salvar vidas. El desarrollo de esto toma alrededor de 4 semanas.

La última fase para el desarrollo de un Web Services es implementar un manejador de solicitudes tipo SOAP, debido a que las solicitudes tipo SOAP se basan en una estructura de 3 componentes: el SOAP Envelope, SOAP Header y el SOAP Body. Esta clase se encarga de desarmar esta estructura para llegar al SOAP Body y poder obtener el cuerpo del mensaje como tal, es decir, el esquema XML con los argumentos y parámetros enviados por el usuario. Esta estructura se puede apreciar en la siguiente figura:

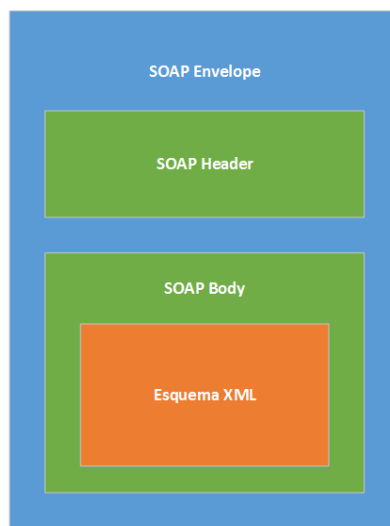


FIGURA 69. ESTRUCTURA SOAP

La utilización de una estructura de mensaje tipo SOAP ofrece un buen desempeño en la atención de varios usuarios, además, de poder ser integrado en el App Engine de Google. Para poder tener un manejador de SOAP que funcione de manera eficiente es necesario tener inicialmente una clase de tipo servlet que pueda atender las solicitudes http, ya que los request generados por los usuarios llegan vía web, por lo tanto se hace indispensable tener una clase que reciba esta solicitud a través de un método `doPost()`. Una vez recibida la solicitud, esta se descompone

según los encabezados de la misma, para así, poder leer el mensaje de tipo SOAP a partir de la petición del cliente. Una vez obtenido este mensaje es compartido a la clase SOAPHandler o manejador SOAP, dicha clase, se encargará de interpretar la solicitud SOAP y trabajar de la mano con el adaptador para brindar una respuesta al usuario. En el manejador SOAP se recibe el mensaje en una estructura SOAP, como la descrita previamente, luego se debe obtener el cuerpo de la solicitud SOAP, que es donde se encuentra realmente la información, ya que en el cuerpo está el esquema XML con los parámetros enviados por el usuario, es decir, se elimina la información que no es requerida para conocer el método demandado por el usuario y los parámetros necesarios para completar la tarea.

Cuando se tiene el cuerpo del SOAP se evalúan los elementos del SOAP, los cuales son el namespace y el nombre del método demandado y con esta información poder decidir qué tarea ejecutar y depurar el esquema XML del resto del mensaje, para ser enviado a la clase adaptador, además, se verifica si la solicitud está estructurada de manera correcta. En caso de existir algún error el manejador envía una excepción al cliente, por medio de la clase que hereda de HttpServlet. Si es posible atender la solicitud, esta clase se encarga de sacar toda la estructura del XML con los argumentos, pasarla al adaptador para que pueda generar la respuesta. El manejador SOAP espera la respuesta del adaptador con el esquema XML, el esquema se agrega a un SOAPBody de respuesta y se genera un nuevo SOAP Message para responder la solicitud. La respuesta es construida por el manejador SOAP y es enviada como una respuesta Web a las aplicaciones a través de la clase que HttpServlet. De esta manera se maneja una solicitud tipo SOAP. Esta fase tomó un tiempo aproximado de 2 semanas para terminar su desarrollo. El proceso de implementación de esta última fase puede apreciarse de una manera más clara en el siguiente diagrama:

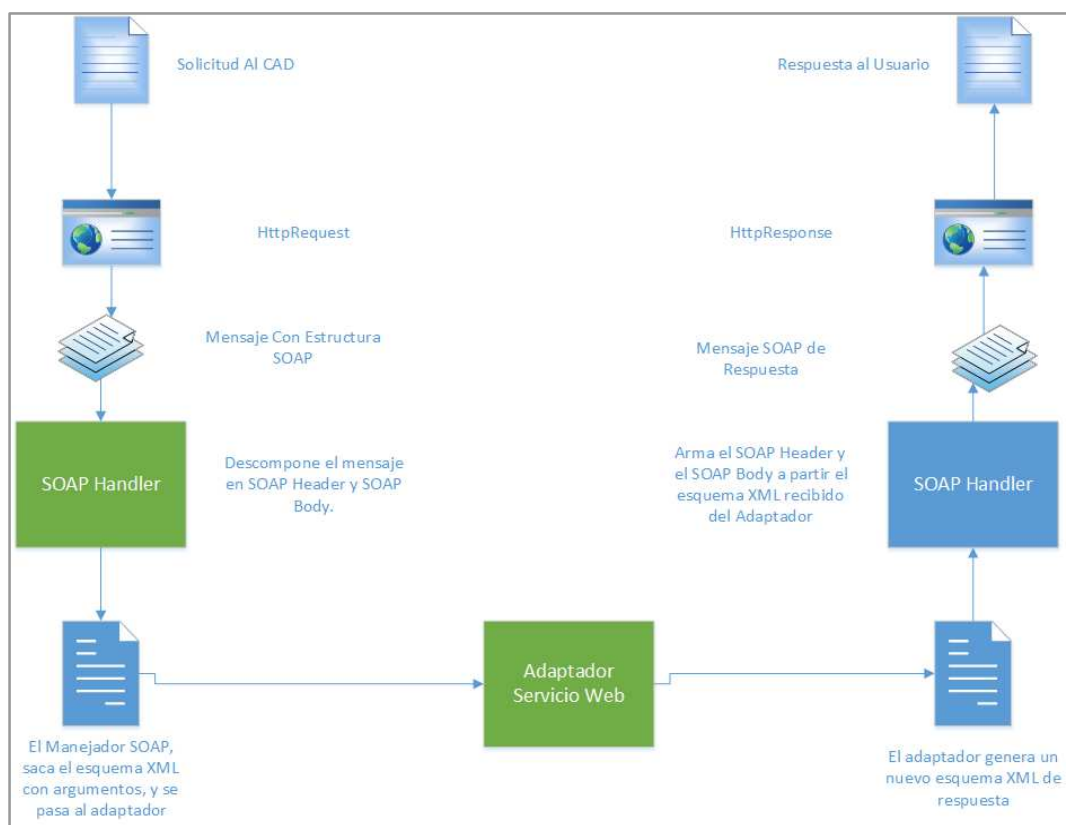


FIGURA 70. COMUNICACIÓN SOAP

En conclusión el desarrollo de un servicio web se divide en 5 fases, cada una orientada a cumplir una tarea específica, pero deben trabajar todas en conjunto para brindar una solución al usuario por medio de una solicitud y una respuesta web, además, al manejar mensajes de tipo SOAP encapsulados dentro de la solicitud se logra tener una estructura firme a la hora de atender una solicitud y así el servidor puede responder a múltiples solicitudes de manera simultánea sin mezclar los datos. Adicionalmente el manejo de un esquema XML permite obtener la información necesaria para ejecutar los diferentes métodos y poder organizar la respuesta, para que las comunicaciones tengan un orden que vaya de la mano con el esquema funcional planteado y, así, con la ayuda de la clase adaptador y los métodos Web poder ofrecer una aplicación robusta y eficiente a los usuarios que demandan los servicios al servidor con buenos tiempos de respuesta, que a la final se ve como un aumento en el performance del servicio.

7.13.2 SERVICIO WEB USUARIO.

Este Servicio web está orientado a manejar la información que será registrada y modificada de los usuarios finales con la aplicación móvil de solicitud de atención. Fue desarrollado en 4 etapas: la primera es la declaración de la clase usuario, la segunda fase es la generación del servicio web y el archivo .wsdl, la tercera es el manejo de los adaptadores de respuesta y la última se centra en el manejo de las solicitudes tipo SOAP. Aquí no se contempla la fase de implementación del Manejador de la Base de Datos, debido a que ya ha sido implementado en el servicio web de almacenamiento.

El propósito de este servicio web es garantizar que las solicitudes de los usuarios finales puedan ser realizadas de manera ordenada y sin afectar las solicitudes de otros usuarios que se estén generando al CAD de manera concurrente, realizando una serie de procedimientos y tareas, debido a las cuales es posible garantizar el buen funcionamiento del sistema con un entorno de comunicación estable, seguro y concurrente. Este servicio se encarga de atender las demandas de los usuarios finales que solicitan la atención de una emergencia.

Las funciones que tiene este servicio se basan en la atención de solicitudes de ambulancias para atender una emergencia médica de 3 tipos: Doméstica, Accidente de tránsito u otro, búsqueda, brindar información de la ambulancia en camino, información de las instituciones cercanas y almacenar una calificación del sistema SIEF. La principal utilidad de este servicio es atender las solicitudes de atención de una emergencia para un usuario, ofrecer una ambulancia cercana al lugar del evento para atender la emergencia, ofreciendo la información de la ambulancia y el tiempo estimado de llegada al lugar, adicionalmente, presentar la información de las instituciones médicas cercanas. De esta manera se ofrece un servicio capaz de cumplir con los requerimientos funcionales planteados para este agente del sistema.

Como se menciona en la introducción de esta sección del documento, el desarrollo de este servicio web se elabora en 4 etapas principales, cada una de ellas será descrita para dar claridad sobre el proceso de implementación del servicio ofrecido. A continuación se muestra un diagrama que ilustra el proceso general:

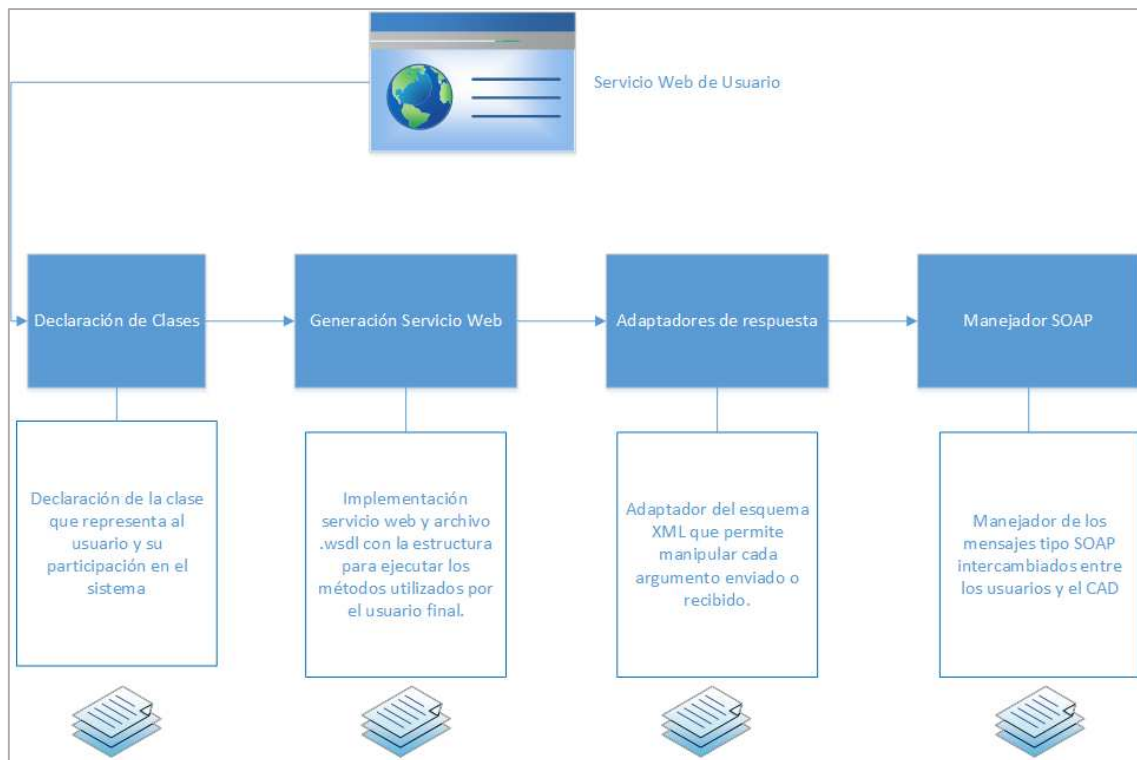


FIGURA 71. MUESTRA GENERAL PROCESO

La primera fase, se centra en declarar una clase para la atención de la solicitud del usuario, ya que esta solicitud hace referencia a la atención de una emergencia en un lugar de la ciudad de Cali y para poder manejarla y darle respuesta al usuario, resulta importante tener una clase que represente dicha solicitud, permitiendo así, el manejo de información que se intercambia entre un usuario y una ambulancia. Es importante definir bien esta clase debido a que representan lógicamente la aplicación del usuario móvil y sus atributos y métodos nos permiten cumplir con los requerimientos funcionales del SIEF. Esta etapa se implementó utilizando Java y su desarrollo tomó un tiempo aproximado de 1 día.

La segunda fase de este desarrollo se basa en el componente dedicado a ofrecer un método que pueda brindar una solución a cada solicitud de los usuarios, referente a la solicitud de atención de una emergencia, es decir, el componente del servicio web encargado de entender que el cliente desea buscar una ambulancia para un traslado a una institución médica, informarle de la ambulancia y el tiempo estimado de llegada de esta. Al igual que en el desarrollo del servicio web de almacenamiento, este es el núcleo del desarrollo de un servicio web ya que los métodos descritos en esta clase, serán los llamados por los dispositivos móviles y los que realicen la secuencia de pasos necesarios para dar respuesta a la solicitud de los usuarios. El proceso de desarrollo de esta etapa se realizó 3 etapas: los métodos de atención de nueva solicitud, métodos de búsqueda de ambulancias cercanas y despliegue de información y métodos de instituciones cercanas. En el diagrama 66 se puede apreciar mejor la funcionalidad de esta tercera fase y el propósito de cada uno de los grupos de métodos definidos:

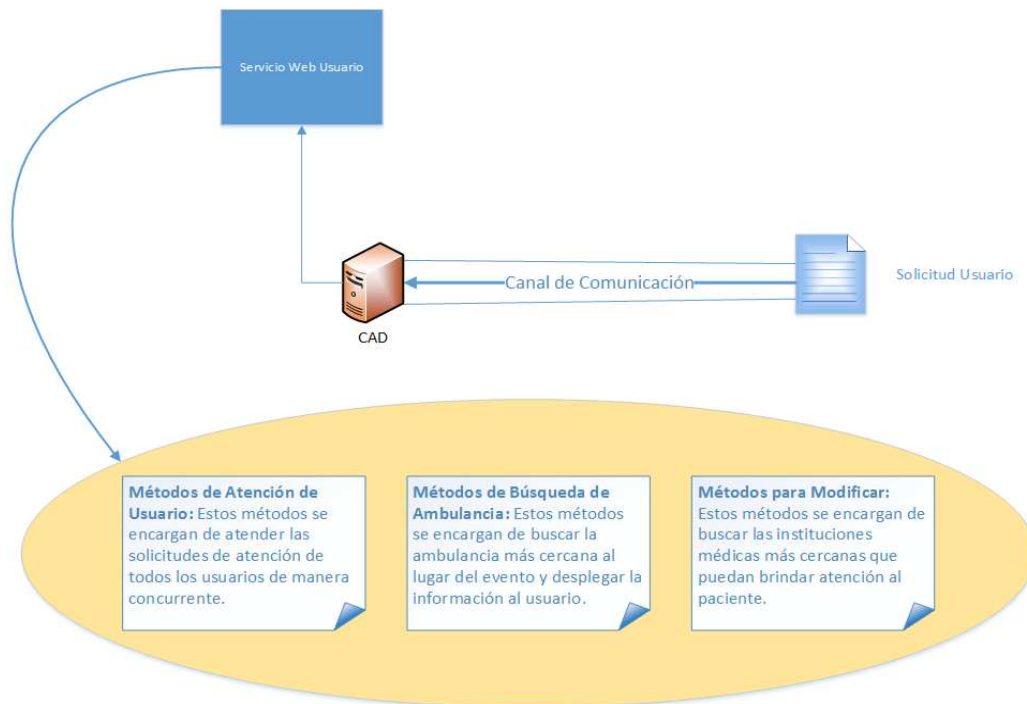


FIGURA 72. FUNCIONALIDAD MÉTODOS USUARIOS.

Pero, al igual que con el servicio Web del almacenamiento, definir una clase con los métodos del servicio Web no es suficiente, es necesario tener un archivo .wsdl que describe la estructura y el esquema del servicio web y su generación se realiza usando los mismos procesos descritos para el servicio web previo, se utiliza el ws-gen, se debe marcar la clase como @WebService y los métodos como @WebMethod. Igualmente, se debe modificar la sección de SOAP address, para poder resolver las solicitudes con el servicio web adecuado. Esta segunda Fase tomó un tiempo aproximado de 2 semanas.

La tercera etapa en la generación de un servicio web funcional, es la generación de un adaptador para el servicio web cuyo objetivo es tomar los argumentos de una solicitud tipo SOAP, obtener su contenido a partir de un esquema XML y resolver las demandas del usuario a la hora de solicitar la atención de una emergencia, ver la información de la ambulancia y el tiempo estimado para llegar a su destino. Al igual que el adaptador del Servicio de Web de almacenamiento se declaran una serie de clases extras de tipo Javax.xml, para poder manejar la estructura de un archivo XML, obtener los argumentos y parámetros de la solicitud enviada y comunicar la respuesta al manejador SOAP utilizando el mismo esquema. Este proceso es igual que el descrito previamente, ya que se crean 2 clases por método definido en la clase núcleo del servicio web, es un trabajo realizado con gusto a pesar de tomar alrededor de 4 semanas su desarrollo.

La última fase para el desarrollo de un servicio web para el usuario es implementar un manejador de solicitudes tipo SOAP, el cual necesita tener inicialmente una clase de tipo servlet que pueda atender las solicitudes http, ya que los request generados por los usuarios llegan vía web, por lo tanto se hace indispensable tener una clase que reciba esta solicitud a través de un método doPost(). Los procedimientos desarrollados en esta clase son iguales que los descritos en la sección del servicio web de almacenamiento, ya que el manejador SOAP recibe el mensaje

en una estructura SOAP, luego se debe obtener el cuerpo de la solicitud SOAP, que es donde se encuentra realmente la información, ya que en el cuerpo está el esquema XML con los parámetros enviados por el usuario. Este esquema es enviado al Adaptador, el cual se comunica con la clase del servicio Web, espera la respuesta y genera el esquema XML para que el manejador SOAP, vuelva a formar la estructura del mensaje y la respuesta sea enviada al usuario o la ambulancia, según sea el caso. De esta manera se maneja una solicitud tipo SOAP. Esta fase tomó un tiempo aproximado de 2 semanas para terminar su desarrollo.

7.13.3 SERVICIO WEB PARAMÉDICO

Este Servicio Web está orientado a manejar la información que será registrada y modifica de los paramédicos o médicos responsables de los traslados hechos por una ambulancia, además, de tener la información de las interacciones hechas con los demás agentes del sistema. Fue desarrollado en 3 etapas: la primera se centra en la generación del servicio web y el archivo .wsdl, el segundo es el manejo de los adaptadores de respuesta y la última se centra en el manejo de las solicitudes tipo SOAP. Aquí no se contempla la fase de implementación del Manejador de la Base de Datos y la clase para el manejo de la información del paramédico debido a que ya ha sido implementado en el Web Service Storage.

El propósito de este servicio web es garantizar que el paramédico responsable del manejo de la ambulancia pueda atender una solicitud de un usuario, establecer comunicación con una institución IPS, documentar al paciente, solicitar el historial de traslados, entre otros. Estos procesos se ejecutan de una manera ordenada, sin afectar las interacciones que otros paramédicos están generando al CAD de manera concurrente, así, es posible garantizar el buen funcionamiento del sistema con un entorno de comunicación estable, seguro y concurrente. Este servicio se encarga de atender las demandas de los paramédicos quienes realizan los traslados, por lo que es un servicio de crucial importancia en el sistema SIEF.

Las funciones que tiene este servicio se basan en la brindar una forma sencilla de administrar los traslados de una ambulancia, incluyendo el historial de traslados realizados o nuevos traslados. Además, de brindar la posibilidad de atender las solicitudes de atención de emergencias generadas por los usuarios o instituciones médicas registradas en el sistema. Otra utilidad es poder modificar el estado de la ambulancia de disponible a ocupado y modificar información básica del paramédico como la contraseña. De esta manera se ofrece un servicio capaz de cumplir con los requerimientos funcionales planteados para este agente del sistema.

Como se menciona en la introducción de esta sección del documento, el desarrollo de este servicio web se elabora en 3 etapas principales, cada una de ellas será descrita para dar claridad sobre el proceso de implementación del servicio ofrecido. Se debe aclarar que en la implementación de este servicio web no se tiene en cuenta el desarrollo de una fase centrada en el manejo de los registros con la base de datos, ni la declaración de una clase que representa lógicamente al paramédico y la ambulancia en el sistema, debido a que estos procedimientos ya fueron realizados en la implementación del servicio web de almacenamiento. A continuación se muestra un diagrama que ilustra el proceso general:

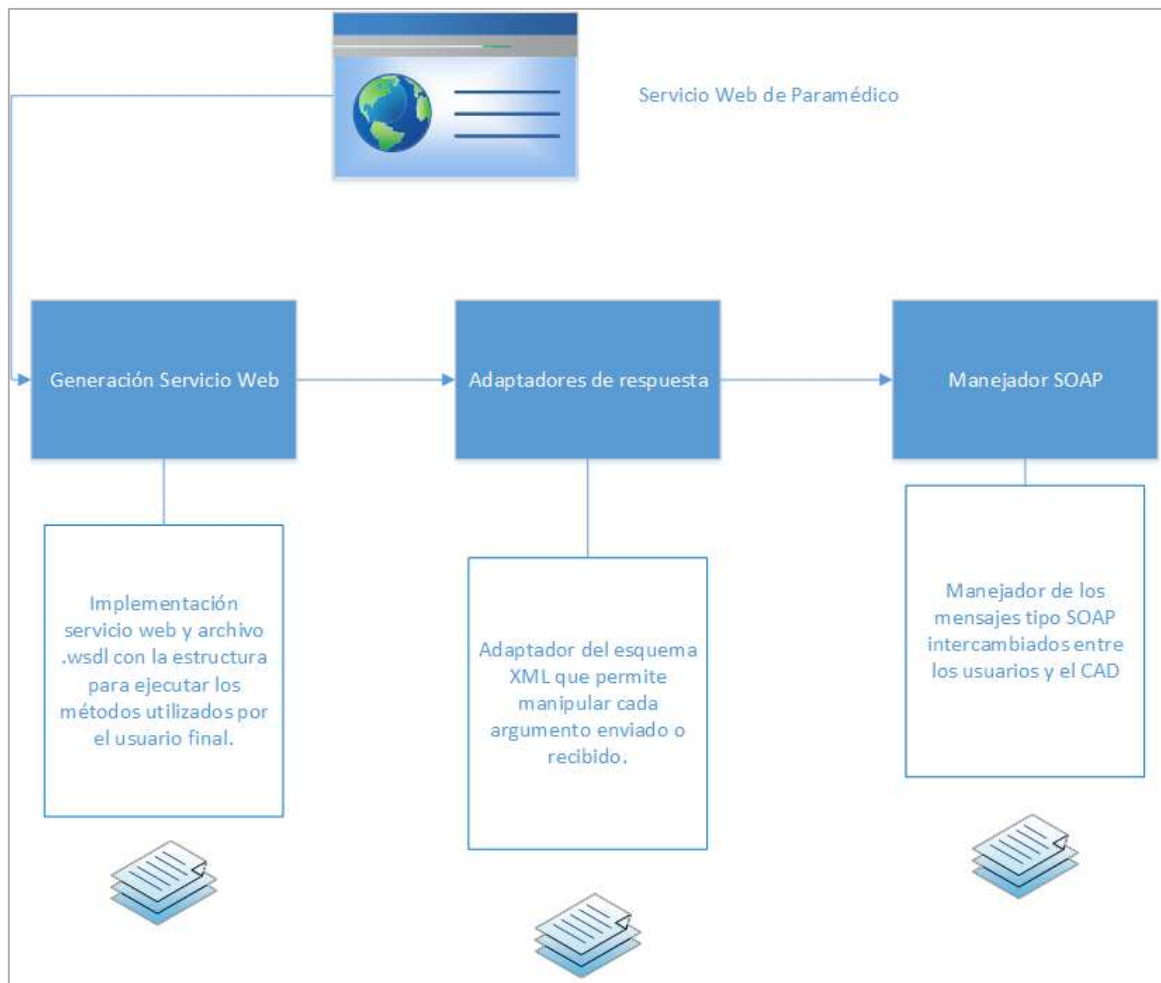


FIGURA 73. PROCESO GENERAL SERVICIO WEB PARAMÉDICO

La primera fase de este desarrollo se basa en la clase dedicada a tener los métodos web que son ejecutados para atender una solicitud de un usuario, establecer comunicación con una institución IPS, documentar al paciente, solicitar el historial de traslados, entre otros. Este servicio web tiene como principal tarea resolver todos los métodos, transacciones y operaciones usadas por los paramédicos para garantizar que su participación en el sistema sea efectuada de la mejor manera, además, de comunicarse con el Manejador de la Base de Datos para guardar los registros de las funciones cumplidas, tales como traslados nuevos, historial de ubicaciones, documentación de un paciente, entre otras. El proceso de desarrollo de esta etapa se realizó 4 etapas: los métodos de atención de nueva solicitud, métodos de historial de traslado, métodos de nuevos traslados y documentación paciente y métodos de modificar estado y perfil. En el siguiente diagrama se puede apreciar mejor la funcionalidad de esta tercera fase y el propósito de cada uno de los grupos de métodos definidos:

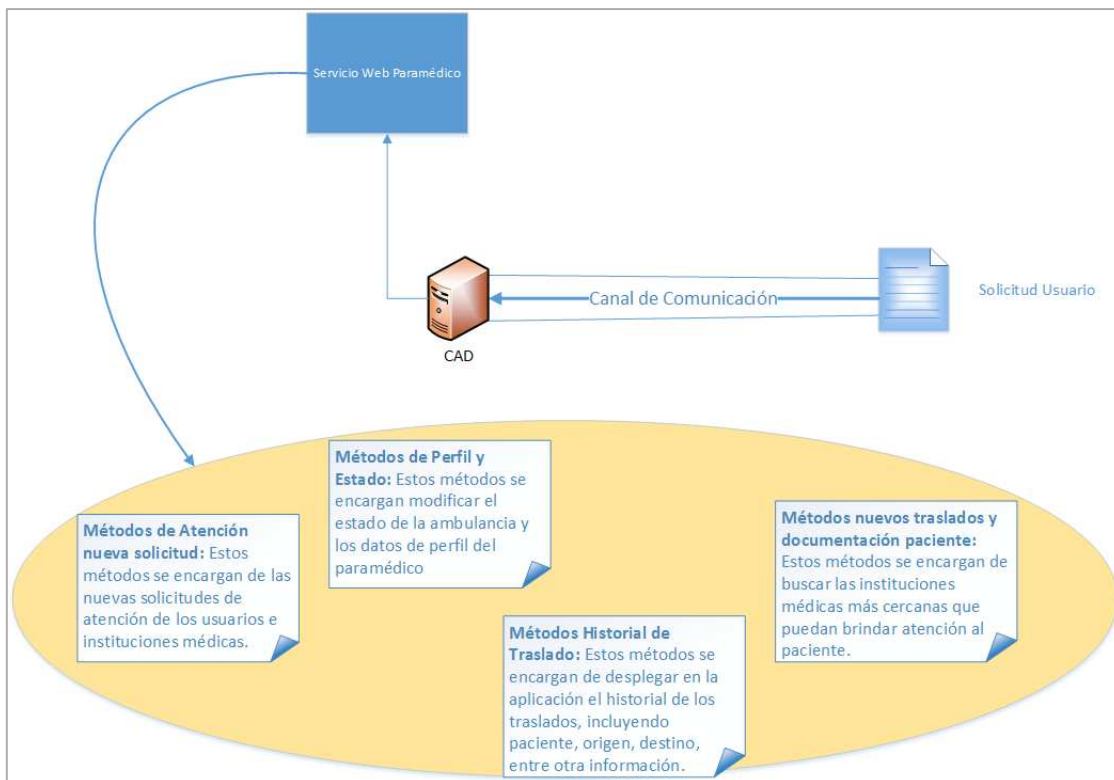


FIGURA 74. MÉTODOS SERVICIO WEB PARAMÉDICO

El funcionamiento de esta clase en general es tomar una serie de parámetros según el objetivo del método y comunicar la ambulancia con los demás agentes del sistema como médicos urgenciólogos, vigilantes y personal de admisiones, adicionalmente, de brindar la posibilidad de tener funciones para actualizar información, ver historial de traslados y documentar un paciente. De esta manera es posible brindar una atención de calidad y organizar las comunicaciones y el intercambio de mensajes. De igual manera que con el servicio Web del almacenamiento, definir una clase con los métodos del servicio web no es suficiente, es necesario tener un archivo .wsdl que describe la estructura y el esquema del servicio web, utilizando el wsgen, marcar la clase como @WebService y los métodos como @WebMethod. Igualmente, se debe modificar la sección de SOAP address, para poder resolver las solicitudes con el servicio web adecuado. Esta primera fase tomó un tiempo aproximado de 1 semanas. El tiempo de trabajo se reduce debido a la experiencia adquirida en el desarrollo, por lo que ya se conoce como implementar una clase que representa el servicio web en menos tiempo.

La segunda etapa en la generación de un servicio web funcional, es crear un adaptador para el servicio ofrecido, tal cual como se ha realizado en los servicios descritos previamente. Este adaptador, como se sabe, obtiene el contenido del SOAP y resuelve las demandas de los usuarios a través de la utilización de los argumentos encontrados en un esquema XML y funciona de intermediario entre el servicio web y el manejador del SOAP. Los procedimientos implementados en esta fase son los mismos, ya que se crean 2 clases por método, es un trabajo realizado con gusto a pesar de tomar alrededor de 4 semanas su desarrollo.

La última fase para el desarrollo de un servicio web para el usuario es implementar un manejador de solicitudes tipo SOAP, el cual necesita tener inicialmente una clase de tipo servlet que pueda atender las solicitudes http, ya que los request generados por los usuarios llegan vía web, por lo tanto se hace indispensable tener una clase que reciba esta solicitud a través de un método doPost(). Los procedimientos desarrollados en esta clase son iguales que los descritos en la sección del servicio web de almacenamiento, se obtiene un esquema XML a partir del

mensaje tipo SOAP. Este esquema es enviado al Adaptador, el cual se comunica con la clase del servicio Web, espera la respuesta y genera el esquema XML para que el manejador SOAP, vuelva a formar la estructura del mensaje y la respuesta sea enviada al usuario o la ambulancia, según sea el caso. De esta manera se maneja una solicitud tipo SOAP. Esta fase tomó un tiempo aproximado de 1 semanas para terminar su desarrollo, con esta etapa ocurre algo similar al desarrollo del adaptador, se gana experiencia en la implementación de los Servicios Web descritos previamente.

7.13.4 SERVICIO WEB INSTITUCIÓN MÉDICA

Este Servicio web está orientado a manejar la información que será registrada, modifica e intercambiada entre las partes y los agentes involucrados con la institución médica, es decir, el personal de admisiones, los médicos y el vigilante. Este servicio web se decide implementar en conjunto para responder solicitudes de varios usuarios, debido a las similitudes en varios de los requerimientos de estos tipos de usuarios, requerimientos como: ver historial de traslados a la institución, el recorrido de la ambulancia durante el traslado, el tiempo estimado, entre otra información. El proceso de desarrollo de esta fase se divide en 3 etapas: la primera es el desarrollo de los métodos del servicio web, el cual se centra en el desarrollo de los métodos, el segundo en un adaptador y por último el desarrollo del manejador del mensaje SOAP. Algo adicional que se describe en esta sección, es la descripción del desarrollo de la aplicación web para el manejo de las admisiones.

El propósito de este servicio web es garantizar que los agentes que laboran en una institución médica, puedan demandar una serie de servicios web que permitan el normal desarrollo de cada una de sus tareas según su rol dentro de la institución. De esta manera, un vigilante no podrá demandar la información del paciente, pero el urgenciólogo si, aplicando así, las restricciones necesarias para garantizar el mejor funcionamiento del sistema SIEF. Estos procesos se ejecutan de una manera ordenada, sin afectar las interacciones que otros agentes que están generando al CAD de manera concurrente, así, es posible tener un entorno de comunicación estable, seguro y concurrente. Este servicio se encarga de atender las demandas de los vigilantes, médicos y personal de admisiones, manejando diferentes hilos de procesamiento para no afectar la comunicación entre las diferentes partes, por lo que es un servicio de crucial importancia en el sistema SIEF.

Las funciones que tiene este servicio se basan en brindar soluciones a las solicitudes comunes entre los vigilantes, médicos y personal de admisiones, guardando una restricción de los roles de cada agente, así aunque los tres pueden solicitar información del traslado, no visualizan en sus aplicaciones los mismo detalles. Con este servicio web es posible atender todas las interacciones que se generan en una institución médica, solicitar un nuevo traslado al CAD, visualizar información del traslado (vigilante, Personal Admisiones, Médico), visualizar información del paciente (Médico), principalmente. De esta manera se ofrece un servicio capaz de cumplir con los requerimientos funcionales planteados para este agente del sistema.

Como se menciona en la introducción de esta sección del documento, el desarrollo de este servicio web se elabora en 3 etapas principales, cada una de ellas será descrita para dar claridad sobre el proceso de implementación del servicio ofrecido. Se debe aclarar que en la implementación de este servicio web no se tiene en cuenta el desarrollo de una fase centrada en el manejo de los registros con la base de datos, ni la declaración de una clase que representa lógicamente al paramédico y la ambulancia en el sistema, debido a que estos procedimientos ya fueron realizados en la implementación del servicio web de almacenamiento. A continuación se muestra un diagrama que ilustra el proceso general:

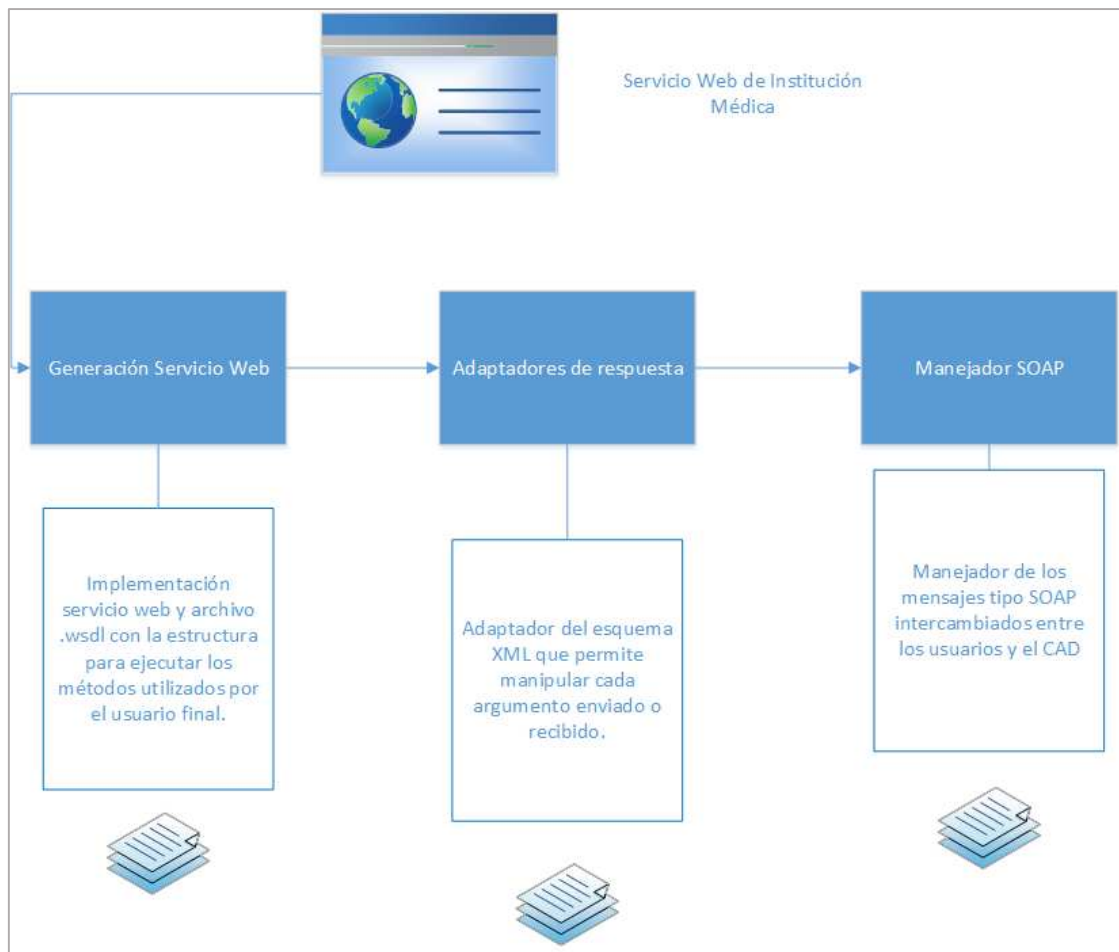


FIGURA 75. PROCESO GENERAL SERVICIO WEB IPS

La primera fase de este desarrollo se basa en la clase dedicada a la atención de las demandas de los usuarios a través de un servicio web, esta atención se hace con métodos que son ejecutados para garantizar que todas las interacciones que se generan en una institución médica sean atendidas y resueltas, este es el servicio web más complejo de desarrollar, debido a que una institución puede generar una nueva solicitud mientras espera un traslado cuyo estado es, en camino o en espera, adicionalmente atiende pacientes; por ende, es el servicio web más crítico y con mayor número de transacciones originadas por diferentes actores.

Se consideró crucial dejar el desarrollo de este servicio web para el final debido a su complejidad y que gracias a la experiencia ganada con el desarrollo de los servicios web anteriores, se reduce la posibilidad de cometer errores en el momento de la implementación y así, desarrollar un servicio mucho más estable en menor tiempo. Este servicio web, además, debe verificar el rol de los usuarios para poder responder solo a las solicitudes que dicho empleado puede solicitar en el sistema, es decir, un vigilante puede consultar el historial de los traslado, pero no puede ver la información del paciente, de esta manera se tiene un servicio más seguro que tiene restricciones según el tipo de usuario. Adicionalmente, el servicio debe trabajar de la mano con el servicio web de almacenamiento para guardar los registros de las funciones cumplidas, tales como solicitudes nuevas, historial de traslados, pacientes atendidos, entre otras. El proceso de desarrollo de esta etapa se realizó 4 etapas: los métodos de restricción de Rol, métodos de información traslado, métodos de nuevos traslados y documentación paciente y métodos de modificar estado y perfil. En el siguiente diagrama se puede apreciar mejor la funcionalidad de esta tercera fase y el propósito de cada uno de los grupos de métodos definidos:

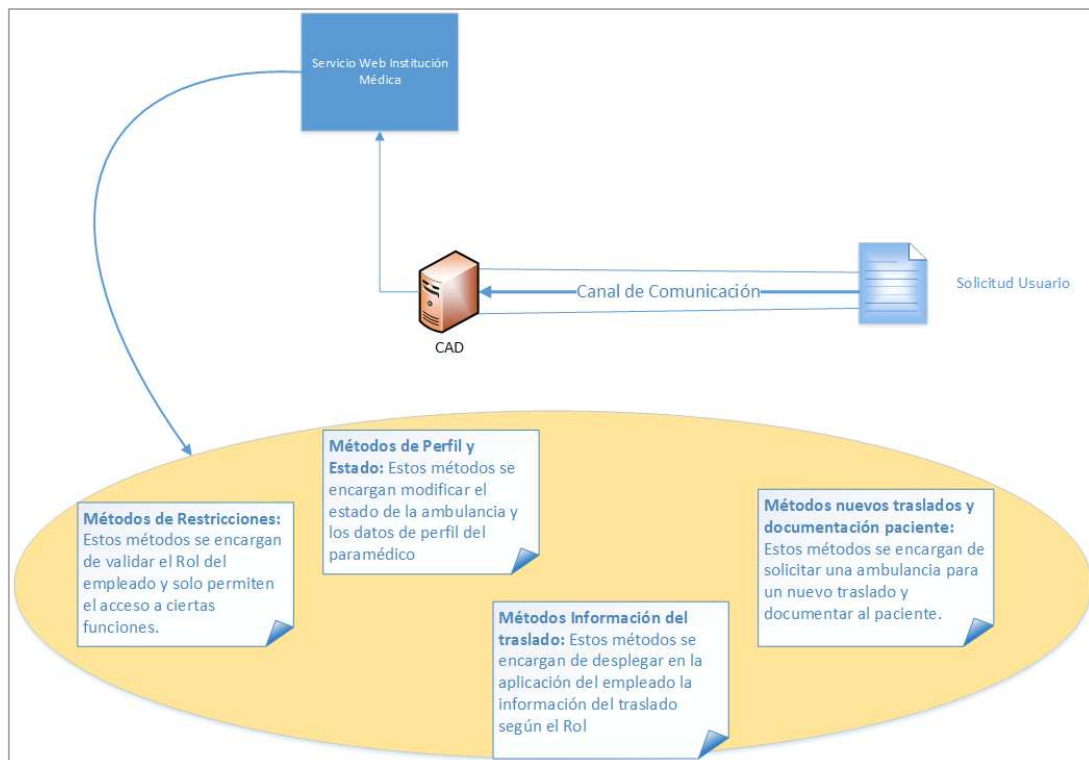


FIGURA 76. MÉTODOS SERVICIO WEB INSTITUCIÓN MÉDICA

El funcionamiento de esta clase en general es tomar una serie de parámetros según el objetivo del método y resolver la solicitud de los agentes de la institución solamente. De esta manera es posible brindar una atención de calidad y organizar las comunicaciones y el intercambio de mensajes. De igual manera que con el servicio Web del storage, definir una clase con los métodos del servicio Web no es suficiente, es necesario tener un archivo .wsdl que describe la estructura y el esquema del servicio web, utilizando el wsgen, marcar la clase como @WebService y los métodos como @WebMethod. Igualmente, se debe modificar la sección de SOAP address, para poder resolver las solicitudes con el servicio web adecuado. Esta primera fase tomó un tiempo aproximado de 3 semanas.

La segunda etapa en la generación de un servicio web funcional, es crear un adaptador para el servicio web, tal cual como se ha realizado en los servicios web descritos previamente. Este adaptador, como sabemos, obtiene el contenido del SOAP y resuelve las demandas de los usuarios a través de la utilización de los argumentos encontrados en un esquema XML y funciona de intermediario entre el servicio web y el manejador del SOAP. Este proceso es igual, ya que se crean 2 clases por método, es un trabajo realizado con gusto a pesar de tomar alrededor de 5 semanas su desarrollo. Este adaptador toma una semana más de desarrollo debido a que en el servicio web se declaran más métodos, ya que es un servicio web para resolver solicitudes de un conjunto de usuarios y no, como en los servicios anteriores, resolver solicitudes de un usuario específico.

La última fase para el desarrollo de un servicio web para atender las solicitudes de los vigilantes, personal de admisiones y médicos urgenciólogos es implementar un manejador de solicitudes tipo SOAP para ofrecer un buen desempeño en la atención de varios usuarios, además, de poder ser integrado en el App Engine de Google. Los procedimientos desarrollados en esta clase son iguales que los descritos en la sección del Servicio web de almacenamiento, ya que se debe tener una clase que reciba la solicitud, la convierta en un SOAP Message para ser pasada al manejador SOAP, el cual verifica que todo esté correcto, extrae el esquema XML y se lo envía al Adaptador,

espera la respuesta y genera la respuesta para ser enviada al cliente o a la ambulancia. De esta manera se maneja una solicitud tipo SOAP. Esta fase tomó un tiempo aproximado de 2 semanas para terminar su desarrollo.

8. REPORTES DE PRUEBA

Las pruebas son necesarias en cualquier tipo de desarrollo si se quiere garantizar la calidad y su salida óptima a producción, constituyen un conjunto de actividades esperadas del software Y existen diferentes tipos, en este trabajo solo se realizaron pruebas funcionales.

El objetivo de la realización de pruebas es proporcionar información objetiva e independiente sobre la calidad del producto tanto al desarrollador como a los usuarios finales. Dichas pruebas pueden ser implementadas en cualquier momento del proceso de desarrollo o cuando éste sea finalizado.

El proceso de prueba puede ser visto como parte del proceso de desarrollo de software o independiente de él. En este último caso, el proceso de prueba no tiene en cuenta la forma en que se realiza el desarrollo para definir las actividades a realizar. En este trabajo, se define un proceso para la prueba funcional de un producto de software, independiente del proceso seguido para su desarrollo. La prueba funcional de un producto de software tiene como objetivo validar cuando el comportamiento observado del producto cumple o no con sus especificaciones. El proceso definido se llama ProTest (Pérez Lamanca, 2006).

Los términos Verificación y Validación suelen usarse indistintamente en algunos contextos. Para las Pruebas del Software ambos términos indican conceptos diferentes por lo tanto es válido aclarar. La definición dada por (IEEE 610.12, 1990), 1990 de Verificación y Validación es la siguiente:

Verificación: proceso de evaluación de un sistema o componente para determinar si un producto de una determinada fase de desarrollo satisface las condiciones impuestas al inicio de la fase.

Validación: proceso de evaluación de un sistema o componente durante o al final del proceso de desarrollo para determinar cuándo se satisfacen los requerimientos especificados. (Boehm , 1984), Usa dos preguntas para clarificar la diferencia entre Verificación y Validación.

Verificación: ¿Estamos elaborando correctamente el producto? **Validación:** ¿Estamos elaborando el producto correcto?

El presente trabajo realiza pruebas funcionales de verificación en los diferentes desarrollos de Back End y Front End.

8.1 AMBIENTE DE PRUEBAS

Para entender mejor el concepto de ambiente de pruebas a continuación se muestran las Sub-etapas de cada ciclo de prueba:

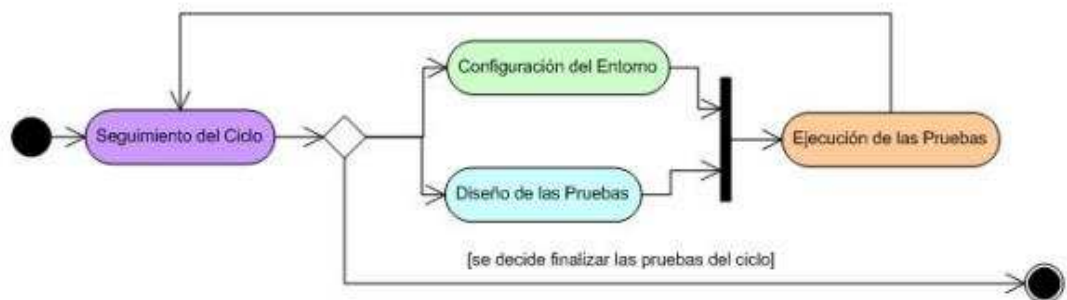


FIGURA 77. ETAPAS DE UN AMBIENTE DE PRUEBAS

Configuración del Entorno: también conocido como ambiente de pruebas, tiene como objetivo configurar el ambiente de pruebas, separando los ambientes de desarrollo y prueba, instalar las herramientas necesarias y el software a probar en la versión correspondiente a cada ciclo de prueba.

En este proyecto se implementó con los emuladores utilizados, copias de código para procesos de calidad y pruebas.

Un caso de prueba (test case) es un conjunto de valores de entrada, condiciones de ejecución, resultados esperados y pos condiciones de ejecución, desarrollados con un objetivo particular o condición de prueba, tal como ejercitar un camino de un programa particular o para verificar que se cumple un requerimiento específico (IEEE 610.12, 1990).

En la prueba funcional las funciones son probadas ingresando posibles entradas y examinando las diferentes salidas. En este tipo de pruebas, la estructura interna del programa raramente es considerada (Kaner, Falk, & Nguyen, 1999).

Para este proyecto se manejaron diferentes componentes de software por lo cual hay lugar a diferentes ambientes de pruebas los cuales han sido separados por aplicación así:

Android Studio: Las pruebas en este software se hacen mediante de emuladores de dispositivos a interés, Smart Phone y Tabletas. Adicional a esto se permite la depuración (debugger) con dispositivos reales si se realiza la instalación de los drivers correspondientes y dicho dispositivo se configura en ambiente de desarrollo.

En este proyecto para las pruebas de SmartPhone en el emulador se creó un dispositivo Galaxy Nexus de 10 pulgadas con el API 21 de android. La instalación de divers y creación de los emuladores lleva un tiempo aproximado de 2 horas.

Las pruebas de navegación se deben hacer en un smartphone real para obtener a través del GPS la ubicación. Para las pruebas de este trabajo se utilizó un Sony z1 compact de 4,3 pulgadas.

Para la simulación de la tableta se utilizó un programa adicional Genymotion el cual se instala como una máquina virtual y requiere del software VirtualBox su instalación completa tarda aproximadamente 5 horas, también para disponer de manera gratuita hay que registrarse y acceder la cuenta una vez instalado el programa, pero las pruebas

de tablet también tienen navegación en sus diferentes ventanas por lo tanto se realizaron pruebas en una tablet Samsung Galaxy Tab 2 de 10,1 pulgadas.

Google Cloud- Web Service

Las pruebas del servidor web se hacen de manera interna con ayuda de la aplicación SOAP UI que permite la conexión con el servidor a través de un SOAP Request, desempeñando el rol de cliente, consumiendo los datos que contiene el servidor.

Las pruebas gráficas y elementales de ingeniería no son incluidas en el archivo adjunto dado su simplicidad.

8.2 CASOS DE PRUEBA

Al realizar pruebas funcionales se debe decidir si la salida observada al ejecutar el programa es la salida esperada.

El desarrollo de SIEF está separado en Back End y Front End por lo tanto se conservará la distinción de los casos de prueba como lo mencionado además de la separación de casos de prueba por cada aplicación.

En total son 27 casos pruebas, que se pueden encontrar en el documento adjunto "*Casos de prueba SIEF.xls*", cada aplicación cuenta con sus pruebas funcionales realizadas, la descripción y la respectiva calificación o resultado obtenido.

TRABAJO A FUTURO

En el futuro este sistema puede interconectar a toda la instituciones médicas de la ciudad e incluso las ubicadas en localidades rurales, además el prototipo funcional puede ser ampliado con funcionalidades que integren equipos médicos presentes en las ambulancias, para realizar el envío de dicha información médica en tiempo real a las instituciones médicas, cambiando el alcance de la nueva integración humano-máquina a internet de las cosas.

La siguiente fase tiene que ver con la unión de este proyecto con el interfaz humano-máquina entregado a las instituciones médicas desde la ambulancia. La integración de estos dos elementos permitirán lograr una comunicación efectiva en un entorno real entre lo que acontece en un ambulancia referente al estado de salud de un paciente y lo que llega a las instituciones médicas como datos relevantes a conocer con antelación.

Se define que el presente trabajo y prototipo funcional aunque tiene proceso de continuación queda de propiedad intelectual los autores.

CONCLUSIONES

En conclusión, podemos decir que el progreso de este proyecto de grado evidencia el arduo trabajo de 2 estudiantes de Ingeniería Telemática para solucionar una problemática presente en la ciudad de Cali que, indudablemente, afecta la prestación de un servicio tan importante para la calidad de vida de las personas como lo es la salud. Una situación que se presenta debido a la falta de una forma de comunicación estable, segura y concurrente entre las ambulancias y las instituciones médicas durante el traslado de un paciente, lo que genera una desinformación con respecto al estado del paciente. Adicionalmente, se identifica otra problemática y es la dificultad a la hora de solicitar una ambulancia para la atención de una emergencia. Basándonos en estos problemas identificados se plantea el objetivo general de este proyecto, aportar una solución para la gestión de la solicitud, traslado e interacción entre un paciente, una ambulancia y una institución médica de forma eficiente e integrada por un sistema que se llamó Sistema Integrado de Emergencias Funcional (SIEF).

Así Pues, teniendo como meta dicho objetivo, se inicia un proceso de trabajo de más de un año gracias al cual fue posible afianzar nuestros conocimientos adquiridos durante el programa de pregrado cursado, ya que se aprendió sobre nuevas tecnologías y protocolos de comunicación existentes que ayudan a la comunicación entre los diversos dispositivos presentes en el mercado. Conocimos nuevas herramientas para el desarrollo de sistemas, nuevas APIs que permiten la implementación de funcionalidades extras a nuestras aplicaciones y diferentes elementos de software que permitieron desarrollar el sistema SIEF. Todos estos factores determinan el esfuerzo y lo aplicado durante el proyecto, pero consideramos que el mayor aprendizaje se da al poder formular una solución para un servicio de TI, cumpliendo con diversos estándares, resolviendo un problema específico de la ciudad de Cali y una necesidad sentida de los usuarios, en este caso, los pacientes que sufren día a día la lentitud, la mala gestión y comunicación de los entes de salud.

Otro componente importante considerado en esta propuesta de solución es trabajar en un proyecto multidisciplinar que nos permitió interactuar en varios contextos conocidos y desconocidos como el mundo de la salud y la atención de emergencias médica, apoyarnos en ingenieros de sistemas y diseñadores para lograr cumplir con la propuesta planteada y escuchar nuestro tutor y asesores en cada uno de inconvenientes presentados, nos deja una gran experiencia de aprendizaje para futuros proyectos que se desean implementar. Comprendimos que para lograr tener éxito se necesita trabajar en conjunto con diferentes disciplinas para poder ofrecer un producto de alta calidad que solucione problemas presentados en nuestra ciudad.

Por último, sentimos que obtuvimos un reconocimiento por nuestros pares en ingeniería por nuestra capacidad para formular soluciones infraestructura y servicios de TI, alineadas con los objetivos de una propuesta de solución o una organización, actuando como empleados líderes que pueden asumir la responsabilidad de proyectos con un impacto crucial para una comunidad definida. Además, demostramos nuestra capacidad para adaptarnos a un entorno en constante cambio, a través de estrategias de aprendizaje permanente que posibilitan nuestra capacidad de liderar y gestionar proyectos de telecomunicaciones trabajando en equipos multidisciplinarios y multiculturales, comprometidos con el desarrollo de soluciones usando las tecnologías de información y comunicaciones que contribuyan al bienestar de nuestra sociedad.

REFERENCIAS BIBLIOGRÁFICAS

- avantel. (2014). Acerca de Avantel. Retrieved Marzo, 2014, from <http://www.avantel.co/inf-corporativa-sobre-avantel/acerca-de-avantel.html>
- Boehm , B. (1984). Verifying and Validating Software Requirements and Design Specifications. IEEE Software.
- Castro Canoa, J. A., Cohen Olivella, E., Lineros Montañez, A., & Sánchez Pedraza, R. (2010). Escala NEDOCS para medir congestión en Urgencias: Estudio de validación en Colombia 38. Retrieved from urosario.edu.co website: <http://bit.ly/1jztkaR>
- Colombia hosting. (2014). cloud hosting. Retrieved Marzo, 2014, from <http://bit.ly/1g5jnRa>
- Comité técnico. (2011). Tipología vehicular. Ambulancias de transporte terrestre. 22. Retrieved from acotaph.org website: http://www.acotaph.org/home/archivos/NTC_3729.pdf
- Cuenca, P. (2011). Historia de la Telemedicina. 17. Retrieved from telemedicinajoaquinanez.weebly website: <http://bit.ly/1jssl1C>
- deconceptos.com. (2014). Concepto Hospital. Retrieved Marzo, 2014, from <http://deconceptos.com/ciencias-naturales/hospital>
- Decreto, 4747 C.F.R. § 3 (2007).
- Dirección Sanidad. (2010). Pliego de Requerimientos No. 029. Bogotá.
- eclipse foundation. (2015). *Desktop IDEs*. Obtenido de <http://bit.ly/1bGI6No>
- EMS1. (2014). ePCR. Retrieved Marzo, 2014, from <http://www.ems1.com/ems-products/ePCR-Electronic-Patient-Care-Reporting>
- enticconfio.gov. (2012). La telemedicina, una realidad en Colombia. Retrieved from enticconfio.gov website: <http://bit.ly/1e95fYt>
- Google Inc. (2015). *Android Studio Overview*. Obtenido de <http://bit.ly/1EDa74R>
- Google. (2013). nexus 7. Retrieved Marzo, 2014, from <http://www.google.es/nexus/7/>
- Google Support. (2014). Registro de desarrolladores. Retrieved Marzo, 2014, from <http://bit.ly/OBhrWp>
- IEEE 610.12. (1990). *IEEE Standard Glossary of Software Engineering Terminology Institute of Electrical and Electronics Engineers*. IEEE.

indicemasacorporal.org. (2007). Definición del índice de masa corporal (IMC). Retrieved Mayo, 2014, from <http://bit.ly/1fGD5oo>
ITMS Chile. (2011). Una nueva era en la telemedicina según ITMS. Retrieved Marzo, 2014, from <http://bit.ly/1kitCdW>

Kaner, C., Falk, J., & Nguyen, H. (1999). *Testing Computer Software, 2nd Edition*. Wiley.

Monteagudo, J. L. (2001). El Marco de Desarrollo de la e-Salud en España. Instituto de Salud Carlos III.

Novarum. (2007). Redes Trunking: La otra cara de la Radiotelefonía. Retrieved Marzo, 2014, from <http://novarum.ver.ucc.mx/n2/0105.html>

Oracle. (2015). *workbench*. Obtenido de MySQL.com: <http://bit.ly/1FnotFm>

Pacheco López, A. (2012). *Telesalud*. Cartagena.

Entrevista para identificar situación actual de los paramédicos, Universidad Icesi (2014).

Resolución, 1448 C.F.R. (2006).

Pérez Lamancha, B. (2006). *Proceso de Testing Funcional Independiente*. Montevideo, Uruguay. Obtenido de <http://bit.ly/1FJ9ZzJ>

Salud Madrid. (2011). Medición de signos Neurológicos (escala de Glasgow). 6. Retrieved from madrid.org website: <http://bit.ly/1nevE7V>

Sanidad, M. (2007). *Conceptos de Telemedicina*. Paper presented at the Plan de Telemedicina del INSALUD, Madrid.

Schwaber, K., & Sutherland, J. (2013). The Definitive Guide to Scrum: The Rules of the Game. 16. Retrieved from scrum.org website: <http://bit.ly/1f9Rncz>

Secretaria Salud. (2014). Especificaciones técnicas mínimas obligatorias de los vehículos *Especificaciones técnicas mínimas obligatorias de los vehículos*. Bucaramanga.

SemanaaSemana.com. (2008). Gestograma. Retrieved Mayo, 2014, from <http://bit.ly/1lXzw2>

Shapiro, J. (2001). Elaboración de un presupuesto. Retrieved from civicus.org website: <http://bit.ly/1hZp0Bc>

Telemedicina Colombia. (2013). colombiantelemed. Retrieved Febrero, 2014, from <http://www.colombiantelemed.com>

Tovar López, J. G. U.S Telemedicina. Retrieved from neutron website: <http://neutron.ing.ucv.ve/revista-e/No5/JTovar.htm>

Universidad de Alicante. (2014). Video Streaming. Retrieved Mayo, 2014, from <http://bit.ly/1q1z9TC>

Velandia Escobar, M. L. (2011). La seguridad se los pacientes de urgencias en un ambiente de congestión. Retrieved from encolombia website: <http://bit.ly/NBfjgJ>

webMedicpro. (2013). AmbuPad. Retrieved Marzo, 2014, from <http://www.webmedicpro.com/index.html>

Álvarez Leiva, C., & Macías Seda, J. (2001). Triage: generalidades *Puesta al día en urgencias, emergencias y catástrofes* (pp. 125-133).

AGRADECIMIENTOS

Agradecemos al PhD (C). Álvaro Pachón que con gran profesionalismo asesoró nuestro proyecto de investigación.

Un especial agradecimiento al Doctor Jorge Asís,
A la Clínica de Occidente y
Al Paramédico Ember Peña,
por habernos brindado la disponibilidad de su tiempo y sus conocimientos en las diferentes
fases de nuestro trabajo.

Al director de programa Juan Carlos Cuellar y
Al profesor Luis Eduardo Munera,
por su colaboración que directa o indirectamente
han hecho posible la culminación y el éxito de este trabajo.

Por último pero no menos importante,
A nuestras madres por su apoyo incondicional.