UCAB Universidad Católica ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

Plataforma para la comunicación multimedia en tiempo real entre pacientes y profesionales del sector de la salud

Este Jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: DIECISIETE (17) PUNTOS

JURADO EXAMINADOR

Firma:	Firma:	Firma:
Nombre: RAFAEL LARA CAMPOS	Nombre: FANNY HERNANDEZ	Nombre: ANA K FERNANDES

REALIZADO POR	**Marco Antonio Franceschi Trujillo**
TUTOR EMPRESARIAL	**Eilyn Millán**
TUTOR ACADÉMICO	**Fanny Hernández**
FECHA	**Junio 2017**

UCAB Universidad Católica ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

Plataforma para la comunicación multimedia en tiempo real entre pacientes y profesionales del sector de la salud

TRABAJO INSTRUMENTAL DE GRADO

Presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

INGENIERO EN INFORMÁTICA

REALIZADO POR	**Marco Antonio Franceschi Trujillo**
TUTOR EMPRESARIAL	**Eilyn Millán**
TUTOR ACADÉMICO	**Fanny Hernández**
FECHA	**Junio 2017**

DEDICATORIA

Este trabajo instrumental de grado va dedicado a mi abuelo, amigo, padre y modelo a seguir Carlos Augusto Trujillo Trujillo.

Siempre me impulsaste a superarme y perseguir mis sueños sin descanso.

Te agradezco por cada sonrisa, cada enseñanza y hasta cada regaño que me convirtieron quien soy hoy en día.

Vivirás por siempre en mi memoria.

Marco Antonio Franceschi Trujillo

AGRADECIMIENTOS

A mi persona, por perseverar hasta llegar a la meta, por aprender de cada fracaso y volver a levantarme para seguir adelante.

A mi familia, por su apoyo y amor incondicional. A mi papá por creer en mí y mostrarme que la vida es más simple de lo que parece, a mis tíos Liz y Alexis por ser un ejemplo de que el trabajo duro rinde sus frutos, a mi mamá Yiyi por recordarme que siempre debe existir el equilibrio y ser mi mano derecha en todo momento, a mi abuela Avo por siempre mantenerse a mi lado y demostrarme lo que significa la verdadera fortaleza.

A mis amigos, los cuales hoy en día considero mi familia, con los que viví alegrías y tristezas durante mi época universitaria, a mis mosqueteros Eilin y Jonathan por convertirse en mis pilares cuándo más los necesité sin que se dieran cuenta.

A mis compañeros de trabajo, por ejercer la presión suficiente para la elaboración de este trabajo de grado, en especial a Gerardo, Mimia, Mariangela, Alexander y Luis por sus contantes correcciones y observaciones desinteresadas.

Y a todas esas personas que conocí durante estos años que de alguna forma u otra llegaron a mi vida para dejarme una lección, gracias.

Marco Antonio Franceschi Trujillo

ÍNDICE

Dedicatoria i

Agradecimientos ii

Índice iii

Índice de figuras vi

Índice de tablas vii

Sinopsis viii

Capítulo I: Planteamiento del problema 1

I.1 Necesidades de la Empresa 1

I.2 Solución propuesta 2

I.3 Objetivos 4

I.3.1 Objetivo General 4

I.3.2 Objetivos Específicos 4

I.3.2.1 Aporte Funcional 5

I.3.2.2 Aporte Tecnológico 5

I.4 Alcance 6

I.5 Limitaciones 13

I.6 Justificación 14

Capítulo II: Marco Teórico 15

II.1 Modelo OSI 15

II.1.1 TCP 15

II.1.2 UDP 15

II.1.3 NAT 16

II.1.4 webRTC 16

II.1.5 STUN 16

II.1.6 TURN 16
II.1.7 Signaling 17
II.2 Aplicativos Móviles 17
II.2.1 Desarrollo móvil nativo 17
II.2.2 Desarrollo móvil web 18
II.2.3 Desarrollo móvil híbrido 18
II.3 Aplicativos web 18
II.3.1 Navegadores web 18
II.3.2 HTML 19
II.3.3 CSS 19
II.3.4 JavaScript 19
II.3.5 AngularJS 20
II.3.6 Servidores web 20
II.4 Servicios web 20
II.4.1 HTTP 20
II.4.2 REST 21
II.5 ORM 21
II.6 Java 21
II.6.1 JPA 21
II.6.2 JAX-RS 22
II.6 Servidor de aplicaciones 22
II.6.1 Wildfly 22
II.8 RabbitMQ 22
Capítulo III: Marco Metodológico 23
Capítulo IV: Desarrollo 25

IV.1 Introducción 25
IV.2 Análisis 26
IV.3 Diseño 35
IV.4 Planificación 39
IV.5 Sprints 39
IV.5.1 Sprint 1 39
IV.5.2 Sprint 2 44
IV.5.3 Sprint 3 47
IV.5.4 Sprint 4 48
Capítulo V: Resultados 51
Capítulo VI: Conclusiones y Recomendaciones 57
VI.1 Conclusiones 57
VI.2 Recomendaciones 59
Bibliografía 60
Apéndices 63
Apéndice A – Artefactos de la metodología 63
Apéndice B – Artefactos del desarrollo 69
Sprint 1 69
Sprint 2 72
Sprint 3 75
Sprint 4 77

ÍNDICE DE FIGURAS

Figura 1 – Proceso de recepción y atención de llamadas. – Elaboración propia. . 11
Figura 2 – Ciclo de Scrum – Fuente:
https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software) 23
Figura 3 – Interacción de la Plataforma – Elaboración propia 26
Figura 4 – Diagrama de arquitectura de servicios – Elaboración propia. 36
Figura 5 – Arquitectura de la Plataforma – Elaboración propia 38
Figura 6 – Modelo entidad-relación – Elaboración propia. 40
Figura 7 – Pantalla de registro (Aplicativo móvil) 69
Figura 8 – Pantalla de inicio de sesión (Aplicativo móvil) 70
Figura 9 – Pantalla de recuperación de contraseña (Aplicativo móvil) 70
Figura 10 – Menú de navegación (Aplicativo móvil) 71
Figura 11 – Pantalla para administrar datos de cuenta de usuario (Aplicativo móvil) 71
Figura 12 – Vista de inicio de sesión (Portal web) 72
Figura 13 – Pantalla de videoconferencia (Aplicativo móvil) 72
Figura 14 – Vista de lista de espera (Portal web) 73
Figura 15 – Vista de video conferencia en progreso (Portal web) 73
Figura 16 – Pantalla de video conferencia en progreso (Aplicativo móvil) 74
Figura 17 – Pantalla para llamada a call center (Aplicativo móvil) 74
Figura 18 – Pantalla de nuevo chat (Aplicativo móvil) 75
Figura 19 – Pantalla de chat (Aplicativo móvil) 75
Figura 20 – Vista de listar chats (Portal web) 76
Figura 21 – Vista de chat (Portal web) 76
Figura 22 – Proceso de transferencia de archivos desde Aplicativo móvil 77
Figura 23 – Proceso de transferencia de archivos desde Portal web 77

ÍNDICE DE TABLAS

Tabla 1- Alternativas de tecnologías para la comunicación en tiempo real [Elaboración propia]. 30
Tabla 2 – Alternativas de frameworks de desarrollo móvil híbrido [Elaboración propia]. 34
Tabla 3 – Historia de usuario 001 63
Tabla 4 – Historia de usuario 002 64
Tabla 5 – Historia de usuario 003 64
Tabla 6 – Historia de usuario 004 64
Tabla 7 – Historia de usuario 005 64
Tabla 8 – Historia de usuario 006 65
Tabla 9 – Historia de usuario 007 65
Tabla 10 – Historia de usuario 008 65
Tabla 11 – Historia de usuario 009 65
Tabla 12 – Historia de usuario 010 66
Tabla 13 – Historia de usuario 011 66
Tabla 14 – Historia de usuario 012 66
Tabla 15 – Historia de usuario 013 66
Tabla 16 – Historia de usuario 014 66
Tabla 17 – Sprint backlog 1 67
Tabla 18 – Sprint backlog 2 68
Tabla 19 – Sprint backlog 3 68
Tabla 20 – Sprint backlog 4 69

SINOPSIS

El Trabajo Instrumental de Grado (T.I.G.) presentado a continuación, lleva por nombre "Plataforma para la comunicación multimedia en tiempo real entre pacientes y profesionales del sector de la salud", propone una solución de *software* en aras de facilitar la comunicación entre pacientes y doctores, reduciendo los tiempos de espera a ser atendidos o diagnosticados, en vista del déficit de personal médico especializado ocasionado por el alto volumen de emigración del país, que provee la Federación Médica Venezolana (F.M.V.), y la dificultad que acarrea satisfacer la demanda generada por la población.

Se utilizó Scrum como marco de trabajo ágil para la gestión del proyecto, adicional a esto se tomaron en cuenta las etapas de Análisis, Diseño y Planificación previas a la etapa de Desarrollo, que a su vez fue dividida en cuatro (4) iteraciones para la culminación de la plataforma, conformada por un aplicativo móvil, un portal *web* y un motor para la extracción de datos.

El aplicativo móvil abre la posibilidad de realizar videoconferencias, crear conversaciones escritas y permitir el envío de documentos o imágenes a los especialistas del área de la salud con acceso al portal *web* perteneciente a las empresas clientes de Trascend, serán capaces de atender, diagnosticar y orientar a los pacientes.

El motor de extracción de datos tiene como objetivo mantener la información de los beneficiarios o titulares (usuarios del aplicativo móvil) actualizada a través de los archivos enviados por distintas compañías de seguro.

La incorporación de la plataforma obtenida como producto de *software* de este trabajo instrumental de grado, introduce a los clientes de Trascend en el mercado de las aplicaciones móviles en el área de telemedicina, proporcionando nuevos canales de comunicación y mejorando la calidad de servicio a sus clientes.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

I.1 Necesidades de la Empresa

El sector de la salud presta servicio a una gran cantidad de ciudadanos, siendo de vital importancia para los pacientes mantener una comunicación directa y efectiva con los profesionales del campo de la medicina. Entre los clientes de Trascend existen organizaciones dedicadas a la prestación de servicios de asistencia médica, entre los cuales figuran la atención de Emergencias 24 horas al día, Servicios de Ambulancia, Laboratorio médico, Telemedicina, además cuentan con una robusta plataforma integrada que le permite coordinar sus servicios de medicina ocupacional de la organización en las principales ciudades del país.

Según la Federación Médica Venezolana (FMV) más de 10.000 médicos han emigrado del país, lo que genera un déficit en el personal especializado necesario para dar un debido seguimiento o atención a la población, mientras esta cifra sigue en aumento cada vez es más difícil satisfacer la demanda generada por los pacientes.

En aras de mejorar la comunicación entre los pacientes y el personal médico, se plantea la creación de una plataforma con la que podrán interactuar a través de un aplicativo móvil y un portal *web*.

I.2 Solución propuesta

El proyecto "Plataforma para la comunicación multimedia en tiempo real entre pacientes y profesionales del sector de la salud" será llevado a cabo por el personal de Trascend C.A., se contará con un Gerente de proyecto y un líder de proyecto encargados de coordinar y establecer la comunicación con el cliente.

La arquitectura del sistema estará conformada por cuatro capas, 1) el aplicativo móvil, 2) el portal *web*, 3) una capa como fuente de datos y 4) una capa de servicios *web*:

1) Para el aplicativo móvil (*FrontOffice*[1]) se propone utilizar un *framework* para el desarrollo de aplicaciones híbridas, con la finalidad de que sea compatible con los principales sistemas operativos móviles (iOS y Android). Desde el aplicativo los pacientes podrán establecer comunicación con los doctores, mediante conversaciones escritas, transferir archivos, realizar videollamadas y también realizar llamadas al *call center* en dónde serán atendidos por el personal capacitado.

2) Para el portal *web* (*BackOffice*[2]) se plantea utilizar un *framework* basado en MVVM[3], respetando la separación de responsabilidades y permitiendo que la aplicación sea escalable. Los doctores podrán acceder al portal web a través de los computadores de su consultorio usando las credenciales registradas en el Directorio Activo[4] de la empresa. Podrán responder los mensajes escritos de los pacientes, atender videollamadas y decidir cuándo cerrar o no una comunicación establecida con los mismos.

3) La capa de fuente de datos permitirá interactuar con la Base de Datos centralizada en donde se almacenará la información del sistema, realizando las operaciones de creación, actualización, consulta y eliminación de registros.

---

[1] **FrontOffice:** Parte de un sistema de información que proporciona servicio directo a los clientes.
[2] **BackOffice:** Parte de un sistema de información destinada a la gestión de la empresa.
[3] **Model-View-ViewModel:** Modelo-Vista-Modelo de vista, es un patrón de arquitectura de software.
[4] **Directorio Activo:** Servicio establecido en uno o varios servidores, el cual permite administrar inicios de sesión en equipos conectados a la red, como también sus políticas.

4) La capa de servicios *web* consistirá en una interfaz de programación de la aplicación (conocida por sus siglas en inglés API[5]) bajo la arquitectura RESTful[6], funcionando como intermediario para comunicar la capa de fuente de datos con el aplicativo móvil y el portal *web*, a través de peticiones HTTP[7].

Los clientes de Trascend relacionados con la telemedicina se asociarán con empresas aseguradoras, las mismas suministrarán una serie de archivos con la información de sus clientes. La información se enviará a un servidor SFTP[8] perteneciente a la infraestructura del cliente, el cual será instalado y configurado por Trascend. Se desarrollará un mecanismo que permita extraer, validar y procesar los datos, para su posterior almacenamiento en el sistema. Adicionalmente deberá detectar cuándo ocurran cambios en los archivos enviados.

Trascend C.A, empresa venezolana especializada en el desarrollo de soluciones de *software*, ha sido contratada por una serie de empresas dedicadas al ramo de la telemedicina para llevar a cabo este proyecto, que permitirá innovar la forma en que los pacientes se contactan con los profesionales de la salud actualmente.

---

[5] **API:** Conjunto de subrutinas, funciones y procedimientos, con la finalidad de ser utilizado por otro software.
[6] **RESTful:** Estilo de arquitectura utilizado para servicios web usando el protocolo HTTP.
[7] **HTTP:** Protocolo de comunicación, utilizado por defecto en la web para la transferencia de información.
[8] **SFTP:** *SSH File Transfer Protocol*, protocolo de conexión segura con la finalidad de transferir, modificar archivos y carpetas.

I.3 Objetivos

I.3.1 Objetivo General

Diseñar e implementar una plataforma para la comunicación multimedia en tiempo real entre los pacientes y profesionales del sector de la salud basado en un aplicativo móvil y un portal *web*.

I.3.2 Objetivos Específicos

1. Diseñar e implementar el modelo de base de datos para el sistema.

2. Desarrollar una API RESTful para la interoperabilidad de los componentes del sistema.

3. Desarrollar el módulo para la extracción de la información de clientes de empresa aseguradora.

4. Desarrollar módulo de registro de usuario para aplicativo móvil.

5. Desarrollar módulo de autenticación de usuarios.

6. Desarrollar módulo de video conferencia.

7. Desarrollar módulo de *chat*.

8. Desarrollar módulo de transferencia de archivos.

I.3.2.1 Aporte Funcional

9. Rediseñar el proceso de comunicación entre pacientes y doctores utilizado a través de un aplicativo móvil.

I.3.2.2 Aporte Tecnológico

10. Evaluar el uso de *frameworks* de aplicaciones híbridas para la implementación del aplicativo móvil.

11. Evaluar tecnologías disponibles para la comunicación en tiempo real a través de conexiones punto a punto.

I.4 Alcance

1. Diseñar e implementar el modelo de base de datos para el sistema.

Para este módulo se diseñará e implementará un modelo relacional de base de datos, el cual se desempeñará como fuente de datos centralizada del lado del servidor que proveerá tanto el portal *web* como el aplicativo móvil, además de una base de datos local en el teléfono que permitirá almacenar datos de configuración. Se contará con el modelo de Entidad/Relación correspondiente y se usarán los principios se seguridad necesarios para preservar la información sensible.

2. Desarrollar una API RESTful para la interoperabilidad de los componentes del sistema.

Se evaluarán los distintos componentes disponibles en Java[9] para la implementación de la capa de servicios web, usando JSON[10] (*JavaScript Object Notation*) como formato utilizado para el intercambio de datos.

La capa de servicios *web* consistirá en una Interfaz de programación de aplicaciones (API) bajo la arquitectura RESTful, el cual permitirá que los componentes del sistema (aplicativo móvil y el portal *web*) se comuniquen a través de peticiones HTTP.

3. Desarrollar el módulo para la extracción de la información de clientes de empresa aseguradora.

Se requiere un módulo que permita la automatización del proceso extraer e importar la información de los usuarios suministrada por las empresas aseguradoras

---

[9] **Java:** Lenguaje de programación diseñado por Sun Microsystems.
[10] **JSON:** Formato de texto ligero para el intercambio de datos.

para su posterior almacenamiento en la base de datos centralizada utilizada por el sistema. Este módulo se puede dividir en dos sub-módulos:

3.1. **Sub-módulo para la extracción de información de usuarios:** Las compañías de seguros suministrarán la información de los asegurados a un servidor SFTP, en formato .txt o de texto plano. Se deberá implementar un mecanismo que permita detectar cuándo ocurren cambios en los archivos enviados para su oportuna lectura y extracción.

3.2. **Sub-módulo para la importación en la base de datos centralizada:** Debido a que solamente los usuarios activos en alguna de las listas suministradas por las compañías de seguro podrán utilizar el aplicativo móvil, es importante mantener actualizado el estado de los pacientes, así como permitir la incorporación de nuevos usuarios.

4. **Desarrollar módulo de registro de usuario para aplicativo móvil.**

Se propone el diseño e implementación de un módulo que permita completar el proceso de registro de nuevos usuarios del aplicativo móvil. Se tomará en cuenta que la creación del usuario sólo será necesaria si no posee una cuenta ya creada, se realizará un proceso de verificación de datos, se utilizarán los principios de experiencia de usuario (UX) para generar un formulario amigable para dispositivos móviles. Adicionalmente se tendrá que validar si el usuario se encuentra registrado por una compañía de seguros antes de proceder con el proceso de registro.

5. **Desarrollar módulo de autenticación de usuario.**

Se requiere un módulo de autenticación de los usuarios permitiendo limitar o filtrar las acciones que podrán realizar en el sistema, para ello se propone el diseño e implementación de dos sub-módulos, uno para los pacientes registrados en el

aplicativo móvil y otro para los doctores encargados de atender las comunicaciones generadas desde el portal *web*:

***FrontOffice*:** Los pacientes podrán ingresar al sistema mediante un usuario y una clave, para validar la identidad en el aplicativo móvil se incorporará un Token en la petición HTTP. Dentro de este sub-módulo se contempla los procesos necesarios para la recuperación de usuario y clave.

***BackOffice*:** La autenticación de doctores será realizada mediante conexión al directorio activo de la organización, lo que permitirá a los médicos ingresar al sistema mediante el usuario y la clave de su computador, el cual es asignado previamente a cada médico por el departamento de tecnología.

6. Desarrollar módulo de video conferencia.

El módulo de video conferencia se encargará de establecer la comunicación entre los doctores y los pacientes a través del formato de videollamada:

***FrontOffice*:** Permitirá que los pacientes puedan establecer una videollamada con alguno de los médicos disponibles (el cuál será elegido por el sistema dependiendo de quién posea la menor carga de trabajo), deberán hacer uso del micrófono y cámara del dispositivo para comunicarse con el doctor. En caso de que no existan doctores disponibles el paciente podrá optar por seleccionar una opción que lo pondrá en una lista de espera, la misma será atendida a medida que los doctores se desocupen.

***BackOffice*:** Permitirá que los doctores atiendan videollamadas entrantes de parte de los pacientes, en caso de tener una videollamada en curso se generará una lista de espera, que contendrá a los pacientes que estén esperando su turno para ser atendidos según la fecha y hora en que ingresó en la misma, una vez que los

doctores finalicen la comunicación en proceso, podrán visualizar la lista para devolver la videollamada a los pacientes en espera.

7. Desarrollar módulo de *chat*.

Se propone el diseño e implementación de un módulo de *Chat* que se encargará de establecer la comunicación entre los doctores y los pacientes a través de conversaciones escritas:

***FrontOffice*:** Los pacientes podrán establecer un nuevo chat con algún doctor que se encuentre disponible, se tendrá un respaldo local en el dispositivo de las conversaciones establecidas, permitiendo acceder al histórico de las mismas, aunque el paciente se encuentre sin conexión.

***BackOffice*:** Los doctores podrán responder conversaciones escritas establecidas por pacientes, consultar el histórico de las conversaciones realizadas con pacientes y tienen la posibilidad de dar como terminada una conversación.

8. Desarrollar módulo de transferencia de archivos.

Se requiere implementar un módulo que permita el envío, conversión y recepción de archivos para la comunicación entre pacientes y doctores, este módulo se divide en 3 sub-módulos claves para su funcionamiento:

8.1. Sub-módulo para la lectura de archivos: En el caso del aplicativo móvil y el portal *web* se deberá acceder al sistema de archivos, se filtrarán los archivos a enviar dependiendo del formato aceptado, sólo se aceptarán archivos con extensión .pdf e imágenes con extensión .png o .jpg, se cargarán en memoria para su posterior manipulación.

8.2. **Sub-módulo para la conversión de archivos:** En el caso del aplicativo móvil y el portal web se tendrán que convertir los archivos en formato BLOB[11] antes de ser enviados y convertirlos de vuelta en su formato original al ser recibidos por el paciente o doctor.

8.3. **Sub-módulo para la recepción de archivos:** En el caso del aplicativo móvil y el portal web se deberá reconocer el tipo de archivo recibido antes de su conversión, para posteriormente ser almacenado en el dispositivo móvil o computador dependiendo del caso.

Aportes

Funcionales

9. Rediseñar el proceso de comunicación entre pacientes y doctores mediante el uso del aplicativo móvil.

Actualmente la comunicación entre pacientes y doctores suele ser presencial o por vía telefónica. El proceso actual es el siguiente:

1. El paciente llama al *call center* de la empresa del cliente en busca de orientación médica.
2. Un grupo de operadores atienden las llamadas de los pacientes.
3. El operador registrará manualmente la información del paciente en un archivo de excel (su nombre completo, su número de cédula, su edad, la dirección en la que se encuentra, y el motivo de su llamada).
4. Luego el operador deberá contactarse con la empresa de seguros para validar los datos del paciente y el estatus del mismo (si está activo o inactivo).
5. Posteriormente el operador transfiere la llamada a un médico orientador suministrándole los datos del caso atendido, él se encargará de diagnosticar al paciente.

[11] **BLOB:** Colección de datos binarios.

6. Si el caso lo amerita se enviará un equipo médico a la dirección del paciente.

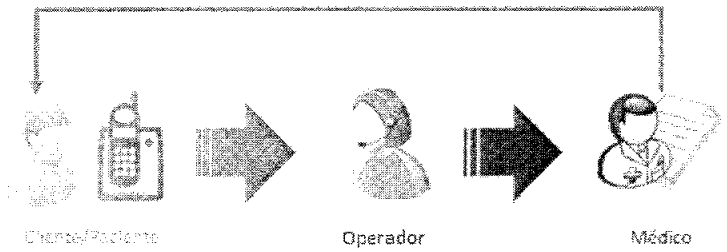

Figura 1 – Proceso de recepción y atención de llamadas. – Elaboración propia.

Se plantea en el nuevo proceso la creación de un aplicativo móvil que acerque al paciente con el especialista de la salud, permitiendo crear comunicaciones efectivas en tiempo real que proporcionen la suficiente información para prevenir emergencias o realizar diagnósticos certeros sin la necesidad de concretar una cita.

La comunicación se llevará a cabo a través de distintos canales, mediante mensajes escritos (*chats*), vía videollamadas y a través de la transferencia de archivos (.pdf, .png y .jpg). Esto brindará una nueva herramienta tecnológica para que los pacientes puedan ponerse en contacto con los doctores y viceversa.

Tecnológicos:

10. Evaluar el uso de "*frameworks*" de aplicaciones híbridas para la implementación del aplicativo móvil.

Se realizará una investigación y evaluación de los distintos "*frameworks*" existentes para la creación de aplicaciones híbridas, esto con la finalidad de que el aplicativo móvil pueda funcionar en distintas plataformas (Android y iOS). Se tendrán en cuenta la capacidad del "*framework*" para acceder a componentes del hardware del dispositivo, tales como cámara y el micrófono.

11. Evaluar tecnologías disponibles para la comunicación en tiempo real a través de conexiones punto a punto.

TCP y UDP son protocolos del nivel de transporte (según el modelo OSI[12]) que permiten la comunicación de datos en distintas aplicaciones. Mientras que TCP es un protocolo orientado a conexión, que garantiza que los datos enviados serán entregados a su destino sin errores y en el mismo orden en que se enviaron, el protocolo UDP es un protocolo no orientado a conexión el cual está enfocado en el envío de mensajes y no tiene confirmación o control de flujo sobre los paquetes enviados. Sobre TCP y UPD existen protocolos de aplicación encargados de establecer y mantener comunicación en tiempo real, entre ellos se encuentran MMS, RTSP, RTP, HTTP, SRTP, cada uno de estos protocolos deberán ser evaluados, conocer beneficios y desventajas para poder elegir el protocolo adecuado para la transmisión de datos.

Se deberá realizar un estudio de las tecnologías existentes que permitan la comunicación entre el aplicativo móvil y el portal *web* en tiempo real, contemplando el uso de conexiones punto a punto. La tecnología elegida pasará por un proceso de evaluación con la finalidad de elegir la que posea la mejor relación fidelidad/latencia, debe ser capaz de permitir llamadas de voz, videollamadas, envío de mensajes escritos y la transferencia de archivos.

También se tomará en cuenta la infraestructura requerida por la tecnología escogida para el funcionamiento eficiente de la plataforma, tomando en cuenta que los pacientes tendrán la posibilidad de comunicarse con doctores desde segmentos de redes diferentes.

[12] **Modelo OSI:** Modelo de referencia para los protocolos de la red de arquitectura en capas.

I.5 Limitaciones

- El desarrollo del proyecto se basará en el marco de trabajo Scrum, debido a que es el utilizado por Trascend.

- Los usuarios de la aplicación móvil deberán estar registrados en alguna empresa aseguradora para poder utilizar la misma.

- Por petición de los clientes de Trascend se limitarán los archivos que podrán ser enviados entre pacientes y doctores, permitiendo sólo archivos con extensión .pdf, .jpg, png.

- Para que los doctores puedan utilizar el portal web deberán estar registrados previamente en el directorio activo de la empresa cliente.

- Debido a que la infraestructura de los clientes está conformada por servidores Linux, se deberán utilizar tecnologías multiplataforma para permitir la instalación de la plataforma en distintos sistemas operativos.

I.6 Justificación

La telemedicina ha ganado importancia en los últimos años debido a la utilización de tecnologías de información y telecomunicaciones, permitiendo que los pacientes se comuniquen con los especialistas de la salud sin necesidad de asistencia presencial. El uso de llamadas, intercambio de mensajes mediante servicios de chat y videoconferencias son algunos de los ejemplos que permiten establecer un enlace entre pacientes y doctores. La inclusión de teléfonos inteligentes (*Smartphones*) ha abierto nuevas oportunidades para mantener contacto con compañeros de trabajo, familiares, amigos y hasta con una amplia variedad de servicios.

Debido al déficit de personal médico especializado ocasionado por la alta cifra de emigración que provee la Federación Médica Venezolana (F.M.V.), es difícil satisfacer la demanda generada por la población. Lo que provoca que los pacientes tengan que esperar largos periodos de tiempo para ser atendidos o diagnosticados.

Tomando en cuenta la problemática planteada, las empresas dedicadas al ramo de la telemedicina han decidido incursionar en el mercado de las aplicaciones móviles y ofrecer un nuevo producto. El cuál permitirá a sus clientes ofrecer nuevos canales de comunicación multimedia entre los usuarios y el equipo de la empresa, facilitando consultas, diagnósticos, y en algunos casos atención de emergencia. A su vez el personal médico contará con un portal *web,* el cual les permitirá prestar sus servicios a los pacientes conectados a través de sus dispositivos móviles.

CAPÍTULO II: MARCO TEÓRICO

Se exponen definiciones sobre herramientas y tecnologías utilizadas en el sistema con la finalidad de aumentar la comprensión del Trabajo Instrumental de Grado (T.I.G.).

II.1 Modelo OSI

Open System Interconnection (OSI o modelo OSI), es un modelo de referencia creado por la Organización Internacional de Normalización (ISO). Ampliamente usado para representar las fases por las que deben pasar los datos para comunicarse desde un dispositivo a otro dentro de una red. Se categoriza a través de siete capas o niveles, pasando desde el más bajo nivel (la capa física) al más alto nivel (la capa de aplicación). Dentro de algunas de las capas se pueden encontrar protocolos que interactúan en el proceso de transmisión y recepción de información.

II.1.1 TCP

Transport Control Protocol (TCP), es un protocolo de red de la capa de transporte del modelo OSI el cuál es orientado a conexión permitiendo que los dispositivos conectados sean capaces de controlar el estado de la transmisión, a su vez garantiza la fiabilidad de los datos al separarlos en paquetes y llevar la secuencia de los paquetes que son entregados mediante un mecanismo de acuse de recibo.

II.1.2 UDP

User Datagram Protocol (UDP), es un protocolo de red de la capa de transporte del modelo OSI, el cual no está orientado a conexión. Esto quiere decir que no proporciona un mecanismo para controlar los datos que son enviados o para la detección de errores en la transmisión de los datos.

II.1.3 NAT

Network Address Translation (NAT), mecanismo que trabaja sobre la capa de red del modelo OSI y tiene como objetivo el comunicar paquetes de datos entre segmentos de red incompatibles, traduciendo direcciones IP[13] entre dispositivos para establecer la ruta de comunicación.

II.1.4 webRTC

Web Real-Time Communications (webRTC), es una tecnología promovida por empresas como Google y la fundación Mozilla, desarrollada por el consorcio W3C[14] para la transmisión vía *streaming*[15] de audio, video e intercambio de datos a través de conexiones punto a punto, sin la necesidad de utilizar *plugins*[16]. Proporciona una API de JavaScript para manejar las comunicaciones establecidas.

II.1.5 STUN

Un servidor STUN (Sesión Transversal de *User Datagram Protocol* [UDP] a través de *Network Address Translators* [NATs]) permite a los clientes NAT (tal como computadores detrás de un *firewall*[17]), configurar llamadas telefónicas a un proveedor VOIP alojado afuera de su red local.

II.1.6 TURN

Un servidor TURN (*Traversal Using Realys around NAT*) es utilizado como mecanismo de *fallback* o respaldo en caso de que falle la conexión establecida a través del servidor STUN. Los servidores TURN consumen una

---

[13] **IP:** Número utilizado para identificar un dispositivo en la red.
[14] **W3C:** World Wide Web o W3C es un consorcio internacional para el desarrollo de estándares web.
[15] **Streaming:** Tecnología utilizada para aligerar la transmisión y reproducción de audio y video.
[16] **Plugins:** Programa de *software* que incrementa las funcionalidades de un programa principal.
[17] **Firewall:** Dispositivo de seguridad de red que permite controlar el tráfico de una red a través de la definición de reglas de bloqueo o acceso.

mayor cantidad de recursos de red y de procesamiento al utilizar el protocolo TCP para la transmisión de los datos.

II.1.7 Signaling

Es el proceso encargado de coordinar la comunicación para establecer los datos necesarios para el intercambio de información entre los clientes.

II.2 Aplicativos Móviles

Una aplicación móvil es un *software* creado con la finalidad de ser instalado en un dispositivo móvil, como puede ser un teléfono inteligente, tableta o incluso un *wearable*[18]. Existen plataformas de distribución ligadas al sistema operativo (Android[19], iOS[20], Windows Mobile[21], entre otros) del dispositivo, en donde se puede encontrar aplicaciones para distintos propósitos.

II.2.1 Desarrollo móvil nativo

Las aplicaciones nativas son desarrolladas para un sistema operativo en específico, utilizando el SDK[22] que ofrece cada plataforma, esto permite un mayor rendimiento y acceso a todas las funcionalidades del dispositivo, como lo son la cámara, el GPS[23], giroscopio, entre otros. Al estar ligadas a un sistema operativo en específico, es necesario invertir mayor tiempo en la curva de aprendizaje y en el desarrollo para que pueda estar disponible en las distintas plataformas de distribución (Google Play, App Store, Windows Store, entre otros).

---

[18] **Wearable:** Accesorios tecnológicos que forman parte del vestuario como relojes o lentes.
[19] **Android:** Sistema operativo móvil de código abierto desarrollado por Google.
[20] **iOS:** Sistema operativo móvil exclusivo de dispositivos Apple desarrollado por Apple Inc.
[21] **Windows Mobile:** Sistema operativo móvil desarrollado por Microsoft.
[22] **SDK:** *Software Development Kit* o kit de desarrollo de software, es un conjunto de herramientas que ayudan a la programación de aplicaciones para un entorno tecnológico particular.
[23] **GPS:** El Sistema de Posicionamiento Global, es un servicio perteneciente a los Estados Unidos, que permite obtener información de posicionamiento y navegación sin suscripción ni cargos alrededor del mundo.

II.2.2 Desarrollo móvil *web*

Las *webapps* agilizan el tiempo de desarrollo al ser realizadas con tecnologías *web* (HTML5, CSS3 y Javascript) de fácil acceso al no necesitar instalación y estar disponibles desde cualquier navegador. Mediante el diseño *web* adaptable pueden adaptar el contenido de la aplicación a distintas resoluciones. Entre sus limitaciones se puede mencionar que no pueden ser accedidas sin conexión de red ni tener control sobre el *hardware*[24] del dispositivo.

II.2.3 Desarrollo móvil híbrido

Las aplicaciones híbridas combinan los beneficios del desarrollo *web* al utilizar tecnologías como HTML5, CSS3 y Javascript donde uno de sus beneficios es acceder a funcionalidades del *hardware* del dispositivo. Poseen un rendimiento levemente inferior al desarrollo nativo, pero provee una mayor reusabilidad de código al poder ser empaquetado y compilado para distintos sistemas operativos.

II.3 Aplicativos *web*

Los aplicativos *web* poseen una arquitectura de cliente/servidor, en dónde en el servidor se encuentra la mayor parte de la lógica de negocios mientras que el cliente se encarga de consumir y mostrar la información recibida por el servidor. En la actualidad el desarrollo de aplicativos *web* se ha incrementado debido a la evolución de estándares como es el caso de HTML5[25] y la mejor compatibilidad de navegadores con las nuevas especificaciones que surgen.

II.3.1 Navegadores *web*

Es una aplicación o *software* utilizado para acceder a sitios *web* o recursos alojados en algún servidor, ya sea a través de Internet o de redes

[24] ***Hardware:*** Compone la parte física de un computador, periféricos y piezas que lo ensamblan.
[25] **HTML5:** Versión de HTML que incorpora nuevas etiquetas y funcionalidades.

locales. Los navegadores *web* son diseñados e implementados por distintas compañías, debido a esto pueden existir diferencias de compatibilidad con las aplicaciones.

II.3.2 HTML

HyperText Markup Language o HTML es un lenguaje de marcas de hipertexto, utilizado como estándar en los sitios *web* para representar la estructura y distribución de los elementos de una página *web*. Es un estándar definido por el consorcio W3C y usado por los navegadores para la visualización de páginas *web*.

II.3.3 CSS

Cascading Style Sheet o CSS es un lenguaje de hojas de estilos diseñado para definir como se mostrará visualmente un documento en pantalla, permite separar el contenido de la presentación de la página *web*. Puede ser utilizado para aplicar estilos a documentos HTML o XML[26].

II.3.4 JavaScript

JavaScript es un lenguaje de programación multiplataforma, orientado a objetos e interpretado, desarrollado a principios de los 90 por Netscape[27] para agregarle dinamismo a los sitios *web*. Posteriormente Netscape entrega la especificación del lenguaje al organismo *European Computer Manufacturers Association* (ECMA) con el fin de estandarizar el lenguaje y desligarlo de cualquier empresa, convirtiéndose en el lenguaje utilizado por defecto en los navegadores *web*.

---

[26] **XML:** *eXtensible Markup Language* es un lenguaje de marcado basado en etiquetas, utilizado para organizar datos.

[27] **Netscape:** Primer navegador *web* comercial.

II.3.5 AngularJS

AngularJS es un framework de JavaScript que utiliza el patrón de ingeniería de *software* MVVM (*Model View View-Model*), su objetivo es la creación de aplicativos *web*, SPA[28] (*Single Page Application*) y también es usado por algunos *frameworks* para el desarrollo de aplicaciones híbridas. Actualmente el proyecto se encuentra mantenido por Google.

II.3.6 Servidores *web*

Un servidor *web* es un *software* o programa encargado de alojar archivos estáticos o aplicaciones *web* y servirlos a través de la red mediante el protocolo HTTP. Funcionan a través de peticiones y respuestas, en dónde un cliente (navegador *web* u otra aplicación) pueden acceder a algún recurso publicado y se entregará una respuesta dependiendo del comportamiento del aplicativo o recurso solicitado.

II.4 Servicios *web*

Es un conjunto de tecnologías o aplicaciones que pueden interoperar en la *web* independientemente de la plataforma o el lenguaje de programación utilizado. Son capaces de intercambiar datos a través de su publicación en la red. Entre los tipos de servicios se encuentran los servicios SOAP, los cuáles transmiten sus datos a través de XML, y los servicios REST, el cual utiliza el formato JSON.

II.4.1 HTTP

Hypertext Transfer Protocol o HTTP es un protocolo de comunicación, desarrollado por el consorcio W3C y la IETF, es utilizado para permitir la transferencia de datos en la *World Wide Web* (WWW). Está orientado a un esquema de petición-respuesta entre un cliente y un servidor.

---

[28] **SPA:** Single Page Application es un tipo de aplicación *web* en dónde todas las pantallas se muestran en la misma página, sin necesidad de recargar el navegador.

II.4.2 REST

Representational State Transfer o REST es una arquitectura de servicios que permite la transferencia de datos entre uno o varios sistemas, usa el protocolo HTTP y sus métodos para acceder a los servicios a través de un URI[29]. Los datos por estándar son transmitidos en formato JSON, aunque también acepta XML.

II.5 ORM

Object Relational Mapping u ORM provee la abstracción necesaria para traducir las entidades generadas en un contexto orientado a objetos a un lenguaje relacional, además de proveer rutinas que permitan interactuar con la base de datos.

II.6 Java

Java es un lenguaje de programación desarrollado por James Gosling y Oracle Corp, de propósito general, utilizado frecuentemente para la creación de aplicaciones *web*, de escritorio, empresariales, entre otras. Su código es compilado y ejecutado desde la JVM (*Java Virtual Machine*) permitiendo la compatibilidad con distintas plataformas y por lo cual es considerado multiplataforma.

II.6.1 JPA

Java Persistence API o JPA proporciona un mecanismo para gestionar la capa de persistencia para aplicaciones desarrolladas con Java EE[30]. Permite a los programadores acceder a un ORM para facilitar el manejo de datos relacionales.

---

[29] **URI:** *Uniform Resource Identifier*, identificador único de un recurso en la red a través de una cadena de caracteres.

[30] **Java EE**: *Java Platform, Enterprise Edition* es la plataforma estándar para el desarrollo de aplicaciones empresariales con Java.

II.6.2 JAX-RS

Java API for RESTful Web Services o JAX-RS proporciona una interfaz de programación para el desarrollo de servicios web RESTful.

II.6 Servidor de aplicaciones

Un servidor de aplicaciones es utilizado para centralizar ciertas aplicaciones en una red de computadores, suelen gestionar las funciones de lógica de negocio y el acceso a datos de la aplicación.

II.6.1 Wildfly

Es un servidor de aplicaciones de código abierto y multiplataforma, para contener aplicaciones empresariales desarrolladas en Java EE. Anteriormente era conocido como JBoss hasta que fue adquirido por la compañía Red Hat, Inc.

II.8 RabbitMQ

Es un *software* de código abierto para la negociación de mensajes, permitiendo que los mensajes a procesar sean encolados y posteriormente desencolados para su tratamiento.

CAPÍTULO III: MARCO METODOLÓGICO

El proyecto utilizó el marco de trabajo Scrum, el cual tiene una estrategia de desarrollo incremental, basándose en una o varias iteraciones para tener como resultado un producto de software. Esta metodología es usada en Trascend C.A. para estimar los tiempos de desarrollo y definir las actividades a realizar por los desarrolladores.

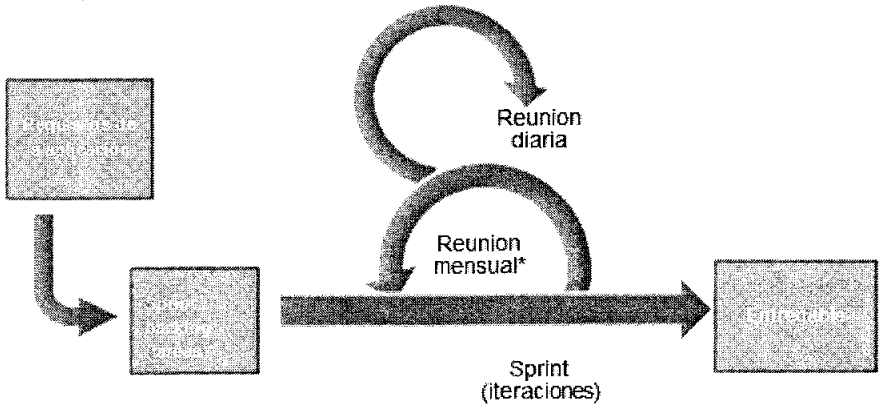

Figura 2 – Ciclo de Scrum – Fuente: https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)

Requisitos: Se establecerán todas las tareas necesarias para cumplir con los requisitos definidos anteriormente junto con la prioridad de las mismas, la lista de tareas o actividades es denominada *Product Backlog*.

***Sprint Backlog*:** Una vez obtenido el *Product Backlog,* se eligen en base a la prioridad de los requerimientos para el cliente una serie de tareas o actividades para ser realizadas en los *sprints* definidos. En esta fase se le asigna una estimación a cada actividad y se determinan el número de sprints.

***Sprint*:** Corresponde al tiempo en el que se van a implementar ciertas tareas o actividades seleccionadas del *Product Backlog*, los *sprints* suelen tener una duración de dos a cuatro semanas. Una vez finalizado el *sprint* se realiza una reunión para medir el desempeño del mismo y evaluar si se lograron todos los

objetivos propuestos.

Reunión mensual: La reunión mensual se establece como mecanismo de control para verificar que se hayan cumplido todas las actividades dentro del tiempo estimado y de ser necesario realizar ajustes antes del comienzo de la siguiente iteración.

Reunión diaria: Las reuniones diarias se utilizan para responder a tres interrogantes dentro del marco de trabajo Scrum "¿Qué actividades se realizaron el día de ayer?", "¿Hubo algún inconveniente?", "¿Qué actividades se van a hacer el día de hoy?".

Entregable: En esta etapa se genera un incremento al producto o sistema final. Al terminar cada una de las iteraciones se realiza una prueba de los requerimientos completados.

Scrum *team*: Contempla los roles desempeñados por los miembros de Trascend C.A. dentro del marco de trabajo Scrum.

CAPÍTULO IV: DESARROLLO

IV.1 Introducción

El desarrollo del sistema fue llevado a cabo a través de cuatro *sprints* con una duración de un mes cada uno, usando como referencia el marco de trabajo Scrum, adicional a esto se tomaron en cuenta varias etapas previas al desarrollo de los *sprints* entre las cuales se evidencian seguidamente:

- *Análisis* en dónde se realizaron reuniones con el cliente y evaluaron tecnologías.
- *Diseño* en dónde se definió el modelo de datos, la arquitectura del sistema y el tipo de usuarios existentes en el mismo.
- *Planificación* en dónde se definió el número de *sprints* y las actividades contenidas en cada uno de ellos.

El comportamiento del sistema está reflejado en la Figura 3, al rediseñar el proceso de comunicación entre pacientes y doctores mediante el uso de un aplicativo móvil y un portal *web*, el cual establece nuevos canales de comunicación para la empresa. Los pacientes pueden comunicarse con el personal médico a través de sus dispositivos móviles, por medio de videoconferencias, mensajes escritos o transferencia de archivos. Mientras que los médicos disponibles podrán atender las solicitudes de los pacientes por medio del portal *web*, en caso de que existan pacientes sin atender podrán verificar la lista de espera una vez se desocupen. El proceso se ve reflejado en la Figura 3.

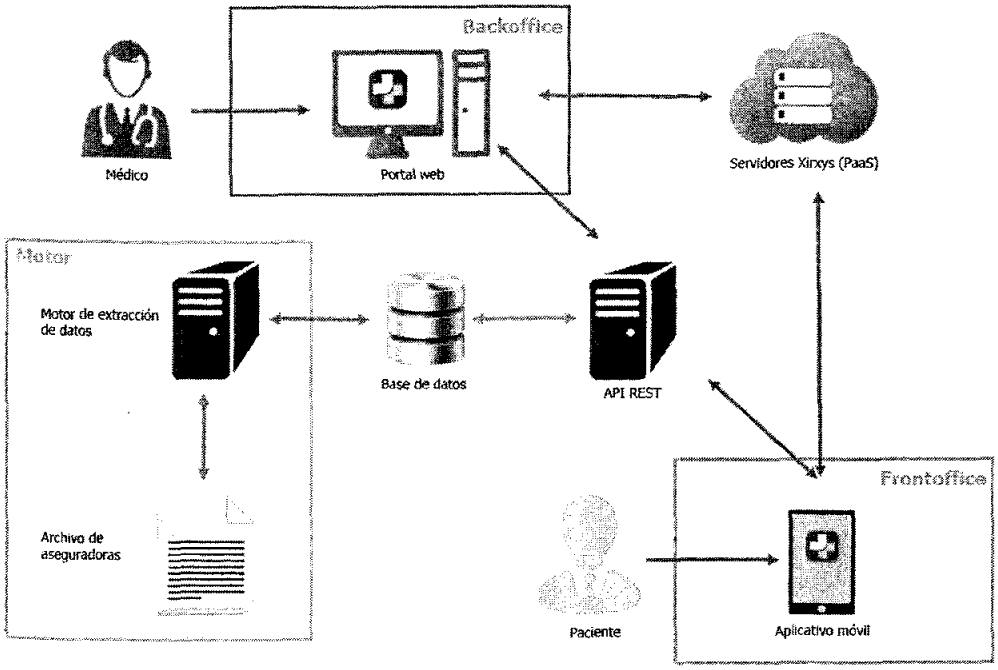

Figura 3 – Interacción de la Plataforma – Elaboración propia

A continuación, se explicará en detalle cada una de las etapas presentes en este desarrollo de *software*:

IV.2 Análisis

Durante la fase de análisis se llevaron a cabo una serie de mesas de trabajo y reuniones con el cliente que permitieron entender los procesos del negocio que maneja la empresa y sus necesidades. Las reuniones dieron como resultado la comprensión del proceso de atención brindado a sus clientes, desde el operador hasta el especialista de la salud. Se realizaron sesiones interdiarias para poder obtener los requerimientos del cliente y proponer soluciones para definir las distintas partes del sistema.

A través de los encuentros con el cliente se tomaron en cuenta las siguientes premisas:

Se evidenció la necesidad de un motor para la extracción de datos, los cuáles serían suministrados por la empresa de seguro a través de un archivo único de texto plano (.txt) con una estructura definida con previo acuerdo entre las aseguradoras y el cliente.

Se definió la necesidad de un portal *web* que permitiría al personal médico de la institución conectarse vía Intranet[31], para poder atender los casos de los pacientes.

Por petición del cliente, el aplicativo móvil a desarrollar debería estar disponible para dispositivos con sistema operativo Android y iOS. Además, se estableció que los usuarios suscritos a alguna de las empresas aseguradoras aliadas puedan acceder al sistema y ponerse en contacto con el personal de la empresa.

Selección de tecnologías

Con la finalidad de seleccionar las tecnologías más adecuadas con base en las necesidades del cliente, se realizó una serie de evaluaciones para poder garantizar la factibilidad del proyecto dentro de los tiempos estimados.

En cada una de las evaluaciones se seleccionó una serie de alternativas gracias a una investigación previa, luego se procedió a dar a conocer cada una de las alternativas, estableciendo criterios para cada una de las evaluaciones.

Para evaluar cada uno de los criterios definidos, cada alternativa dispone de una escala del cero al dos, siendo cero la solución menos factible o recomendable y dos la solución más adecuada o factible. Una vez calculada la sumatoria de todos

[31] **Intranet:** Red informática que utiliza los mismos protocolos que la Internet, pero es destinada a uso interno de una organización o institución.

los criterios por alternativas, se establece la alternativa escogida en base a su valor numérico.

Estudio y selección de las tecnologías disponibles para la comunicación en tiempo real.

Se investigaron diferentes protocolos para la transferencia de datos en tiempo real tomando como premisa que se utilizaría el desarrollo móvil híbrido y la implementación de un portal *web*, por lo que la compatibilidad con los navegadores fue un criterio clave para la etapa de preselección incorporando a *WebSocket* como alternativa y descartando los protocolos:

1) *Microsoft Media Server* (MMS) debido a su poca flexibilidad al ser propietario, y su escasa compatibilidad con aplicaciones *web*.

2) HTTP por sus limitaciones técnicas frente a otras alternativas.

En lo consecuencia de lo descrito anteriormente se realizó el estudio a los siguientes protocolos:

RTP/RTCP

Real-time Transport Protocol (RTP) es un protocolo de transporte en tiempo real para la transmisión de audio y video sobre internet, es altamente usado para las comunicaciones VoIP, siendo considerado como un estándar. Funciona sobre el protocolo UDP y no posee un mecanismo para el control de errores, debido a esto trabaja en conjunto con el protocolo RTCP (*Real Time Control Protocol*) el cual se encarga de verificar la calidad de la transmisión de los paquetes enviados. Existe una extensión del protocolo RTP que incorpora mejoras de seguridad, agregando encriptación, autenticación de mensaje e integridad de los datos denominado *Secure Real Time Protocol* (SRTP) o *Secure Real Time Control Protocol*.

WebSocket

El protocolo WebSocket permite una comunicación *full-duplex*[32] entre un cliente y un servidor a través de un único socket TCP. Es utilizado como alternativa a las distintas técnicas de comunicación en tiempo real, que permite implementar el protocolo HTTP, en dónde mantenía abierto varias conexiones para tener una comunicación bidireccional.

RTSP

Real Time Streaming Protocol (RTSP) es un protocolo no orientado a conexión de la capa de aplicación, destinado a la transmisión de audio y video vía *streaming*, utiliza la combinación de los protocolos TCP y UPD para la distribución de contenido multimedia. Mientras que TCP es usado para controlar la transmisión, el envío de los datos ocurre mediante UDP. Al ser un protocolo no orientado a conexión, se debe mantener una sesión para poder establecer un canal entre el servidor y el cliente.

Criterios de evaluación (0-2 puntos).

- **Estabilidad:** Determina que la conexión esté siempre disponible y sea confiable a la hora de transmitir información.

- **Herramientas de desarrollo:** Determina la cantidad de *frameworks* disponibles para desarrollar usando el protocolo y la documentación adecuada.

- **Fiabilidad:** Determina que los datos transmitidos a través de la red sean confiables y no estén corruptos.

[32] **Full-duplex:** Permite transmitir en ambas direcciones simultáneamente por el mismo canal.

- **Compatibilidad con navegadores:** Determina el soporte nativo de los navegadores para las tecnologías que utilicen el protocolo.

Dado los criterios, se procedió al estudio y selección de las herramientas de comunicación en tiempo real. La Tabla 1 - Alternativas de tecnologías para la comunicación en tiempo real presenta las ponderaciones correspondientes a las tecnologías presentadas a evaluar.

Tabla 1- Alternativas de tecnologías para la comunicación en tiempo real [Elaboración propia].

	RTSP	**Websocket**	**RTP/RTCP**
Estabilidad	2	1	2
Herramientas de desarrollo	1	2	2
Fiabilidad	2	1	2
Compatibilidad con navegadores	0	2	1
	5	6	7

Al totalizar los puntos obtenidos para cada alternativa, se evidencia que la alternativa factible es el protocolo RTP/RTCP con un total de siete puntos, sobre las otras alternativas, *Websocket* con seis puntos y RTSP con cinco puntos.

Según la comparativa reflejada en la Tabla 1 - Alternativas de tecnologías para la comunicación en tiempo real, RTP/RTCP es el protocolo seleccionado para la comunicación en tiempo real, teniendo en cuenta que posee una buena estabilidad de conexión con dos puntos, variedad en soluciones e implementaciones del protocolo con dos puntos y fiabilidad de los datos con dos puntos y en la compatibilidad con navegadores con un punto.

Considerando la elección del protocolo RTP/RTCP para el desarrollo del sistema, se seleccionó webRTC *(Web Real-Time Communications)* como tecnología de comunicación en tiempo real a través de conexiones punto a punto. Promovida por empresas como Google y la fundación Mozilla, desarrollada por el consorcio W3C, webRTC permite la transmisión vía *streaming* de audio y video, además del intercambio de datos y transferencia de archivos sin la necesidad de utilizar *plugins*, aportando la mejor relación fidelidad/latencia; y proporciona una API de JavaScript para manejar las comunicaciones establecidas.

Para comunicar redes en segmentos de red distintos webRTC debe contar en su infraestructura con servidores denominados STUN y TURN para conectar ambos clientes a través del mecanismo de *Signaling.* Por lo que debido a decisión de Trascend se contrató a Xirxys debido a su facilidad de configuración e intereses económicos del cliente, proveedor de *Platform as an Service* (PaaS[33]) suministrando servidores STUN y TURN con alta disponibilidad y dedicados a trabajar con aplicaciones que utilicen la tecnología de comunicación en tiempo real.

Estudio y selección del *framework* de desarrollo móvil híbrido para aplicativo móvil.

Para esta evaluación se realizó una etapa de preselección en donde se estudiaron los *frameworks* de desarrollo móvil híbridos más utilizados, se tomaron en cuenta 3 y se sometió a una comparación a través de distintos criterios, para finalmente definir cuál traería mayor beneficio en la implementación del aplicativo móvil.

[33] **PaaS:** *Platform as an Service*, provee servicios en la nube para suministrar plataformas a desarrolladores.

Criterios de evaluación (0-2 puntos).

- **Compatibilidad:** Determina el correcto funcionamiento e integración de los *Plugins* que dan acceso a funciones del *hardware* del dispositivo, se toma en cuenta el soporte que se le dé a los mismos.

- **Apariencia:** Determina la semejanza que poseen las aplicaciones desarrolladas con el *framework* con aplicaciones nativas, respetando los estilos de cada sistema operativo, permitiendo extender y crear nuevos estilos en base a las especificaciones de cada plataforma.

- **Comunidad:** Determina si posee una comunidad activa que permita el desarrollo constante del *framework* previniendo y solucionando la aparición de errores, y suministrando mejoras.

- **Documentación:** Determina si posee una fuente de recursos, ejemplos o tutoriales que permitan iniciar en los primeros pasos con el desarrollo móvil híbrido, a su vez califica la calidad de la documentación oficial brindada por el *framework*.

La comparación se realizó teniendo en cuenta los siguientes *frameworks*:

Onsen UI

Es un *framework* de código abierto para creación de aplicaciones móviles híbridas, en un principio sólo contaba con estilos nativos para iOS, pero actualmente ha incorporado Material Design[34] para las aplicaciones Android. Utiliza Phonegap/Cordova[35] para acceder a las funcionalidades de *hardware* del dispositivo. Entre sus principales características se destacan:

---

[34] **Material Design:** Guía de estilos utilizada por Google para el diseño de sus aplicaciones.

[35] **Phonegap/Cordova:** Cordova y Phonegap son frameworks que permiten generar aplicaciones multiplataforma a través de una misma base de código utilizando HTML, CSS y Javascript.

- No posee dependencias a un *framework* en específico, optando por la posibilidad de elegir el *framework* de JavaScript de preferencia. Entre las opciones están AngularJS, Angular 2, React, Meteor, etc.
- Provee herramientas como una línea de comandos y una aplicación de escritorio que permiten simplificar tareas, además cuentan con herramientas para depuración, *live-reload*[36], y muchas más.
- Fácil aprendizaje debido a su completa y extensa documentación dado que posee tutoriales, ejemplos y recursos en línea que facilitan el uso y consulta de las funcionalidades del *framework*.

Framework7

Es un *framework* gratuito y de código abierto para el desarrollo móvil híbrido (Phonegap) o aplicaciones *web* destinadas a plataformas iOS o Android. Tiene el objetivo de reducir la curva de aprendizaje al desarrollar el aplicativo móvil como si se tratara de un sitio *web*, sin la necesidad de aprender otro *framework* de JavaScript. Entre sus aspectos más relevantes están:

- Posee animaciones, estilos y gestos táctiles de iOS para que la experiencia sea lo más cercana a su versión nativa. También posee estilos, efectos e interacciones basados en la especificación oficial de Google sobre *Material Design* para aplicaciones Android.
- Posee una gran cantidad de elementos de interfaz de usuario que facilitan la creación de la aplicación como modales, pestañas, sistemas de grillas, formularios, etc.
- Estilos de simple modificación gracias a que están modularizados y divididos en distintos archivos utilizando el preprocesador de CSS Less, facilitando la edición y creación de estilos propios.

[36] **Live-reload:** Monitorea cuándo ocurren cambios en algún archivo, refrescando automáticamente la aplicación.

Ionic

Ionic es un *framework* gratuito y de código abierto, destinado al desarrollo de aplicaciones móviles híbridas. Está construido con SASS[37] y optimizado con AngularJS. Entre sus principales características se encuentran:

- **Alto rendimiento:** Está enfocado en tener un buen rendimiento en dispositivos móviles, gestos táctiles optimizados, aceleraciones de transiciones por *hardware*, entre otras características.
- **Diseño atractivo:** Diseño orientado dependiendo del sistema operativo utilizado, guías de estilos para manejo de grillas, efectos visuales, entre otros.
- **Múltiples *Plugins* nativos:** Aproximadamente más de 120 *plugins* de Codova/Phonegap que permiten comunicarse con funcionalidades nativas del dispositivo, como puede ser Bluetooth, cámara, huella digital, etc.
- **Documentación:** Amplia documentación con ejemplos, demos y tutoriales que muestran cómo crear el primer proyecto hasta la instalación y configuración de los distintas API disponibles.

Se procede al estudio y selección del *framework* de desarrollo móvil híbrido. En la Tabla 2 - Alternativas de *frameworks* de desarrollo móvil híbrido, se muestran los criterios a tomar en cuenta con respecto a cada una de las herramientas estudiadas ponderadas según su factibilidad para cada caso.

Tabla 2 – Alternativas de *frameworks* de desarrollo móvil híbrido [Elaboración propia].

	Onsen UI	**Framework7**	**Ionic**
Compatibilidad	1	1	2
Apariencia	1	2	2
Comunidad	1	1	2
Documentación	2	1	1
	5	5	7

[37] **SASS:** *Syntactically Awesome Style Sheets* es un preprocesador de CSS que permite generar hojas de estilos dinámicas a través del uso de variables, operaciones matemáticas, entre otras.

Al totalizar los puntos obtenidos de cada alternativa, se puede observar que Ionic resulta ser el más factible, al tener siete puntos a diferencia de las alternativas de Onsen UI y Framework7 que obtuvieron cinco puntos cada una.

Ionic ha sido el *framework* seleccionado por obtener una mejor compatibilida, una apariencia adecuada, una comunidad más activa con dos puntos cada criterio, mientras que una documentación suficientemente buena arrojó un resultado de un punto. Tal y como se refleja en la Tabla 2 - Alternativas de *frameworks* de desarrollo móvil híbrido

IV.3 Diseño

Con base en las comunicaciones establecidas con la empresa y las mesas de trabajo se llevó a cabo el levantamiento de información, creando historias de usuario que permiten delimitar las funcionalidades del sistema, teniendo como origen los requerimientos del cliente. En esta etapa se planteó la arquitectura implementada, a su vez se definieron las tecnologías a utilizar con base en la investigación previa en la etapa de análisis sobre la infraestructura del cliente y sus necesidades. Se tomaron en consideración las recomendaciones del arquitecto de *software* de Trascend C.A.

Arquitectura y tecnologías

Debido a las intenciones de la empresa de expandir sus sistemas, se usó una arquitectura basada en capas que favorece su escalabilidad. El sistema está constituido por cuatro capas y un motor de extracción de datos explicados en detalle a continuación:

1) Aplicativo móvil: *FrontOffice* del sistema, permite a los pacientes registrados ponerse en comunicación con el médico. Disponible para dispositivos iOS y Android a través de una aplicación móvil híbrida desarrollada con Ionic y AngularJS.

2) Portal *web*: *Backoffice* del sistema, permite al personal médico de la empresa atender a pacientes a través de una aplicación *web* dentro de la Intranet de la empresa cliente. Se implementó AngularJS para aprovechar la reutilización de código del aplicativo móvil.

3) Capa de datos: Fuente de datos del sistema donde se encarga de todas las transacciones con la base de datos a través del uso de JPA como ORM. Como manejador de base de datos se optó por MySQL[38], por ser un manejador de base de datos multiplataforma y de simple configuración.

4) Capa de servicios *web*: Lógica de negocios del sistema, encargada de interconectar el sistema a través de una API REST utilizando JAX-RS[39] y Java EE 7. La Figura 4 muestra como está diseñada la capa de servicios de la plataforma.

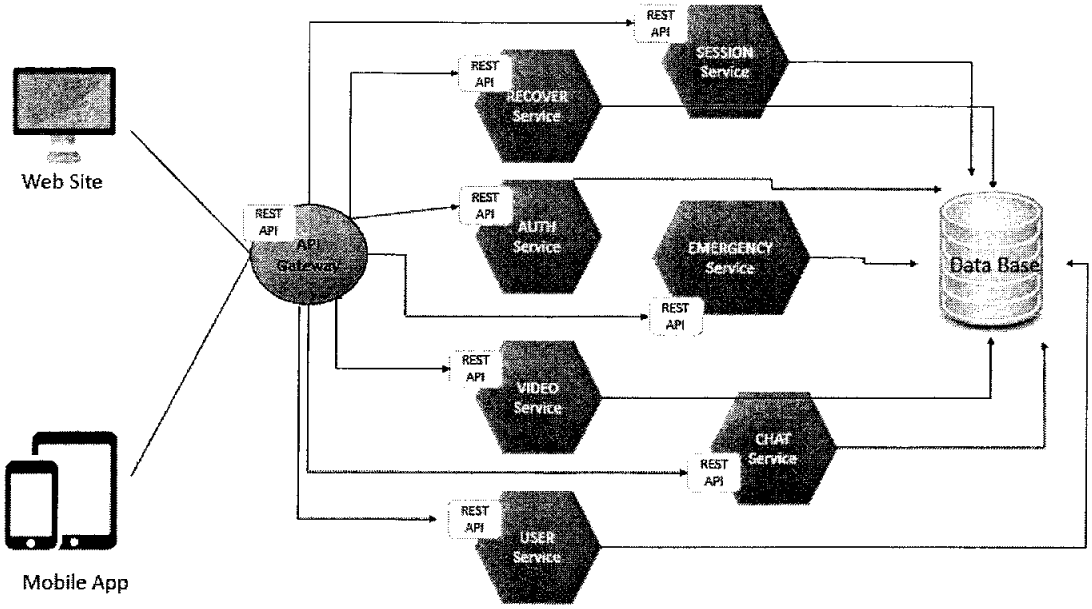

Figura 4 – Diagrama de arquitectura de servicios – Elaboración propia.

[38] **MySQL:** Base de datos de código abierto perteneciente a Oracle Corporation, utilizada como principal opción en la web.
[39] **JAX-RS:** Es un API de Java para el desarrollo de servicios web RESTful.

5) Motor de extracción de datos: Servicio o demonio que está a la escucha de la recepción de nuevos archivos suministrados por las aseguradoras, para su procesamiento y persistencia en la base de datos. En su elaboración se utilizó la plataforma Edición Estándar de Java, Java SE 8, debido a su API para el manejo de hilos.

Para la comunicación en tiempo real se escogió PeerJS[40] como librería la cual facilita la implementación de webRTC a través de un API de JavaScript orientado a eventos.

En la Figura 5, se presencia gráficamente la interacción y comunicación de los nodos descritos anteriormente, visualizando la arquitectura del sistema completo.

[40] **PeerJS:** Es una librería que envuelve la implementación de webRTC, brindando una serie de métodos que facilitan la configuración e implementación de conexiones punto a punto.

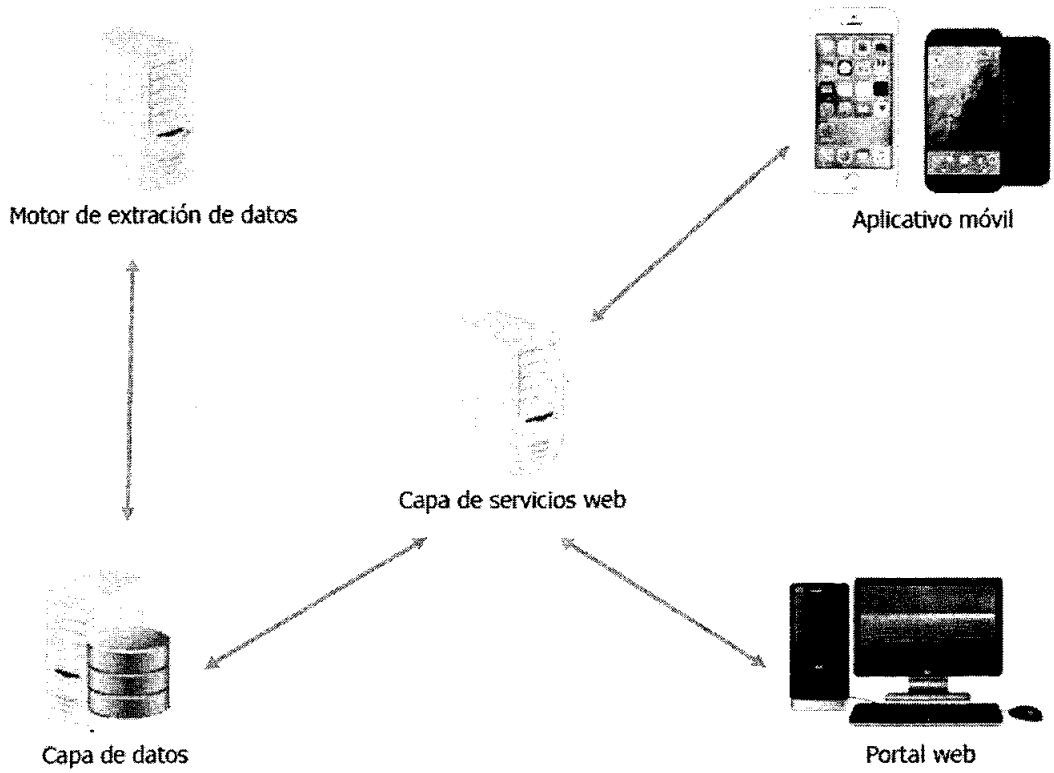

Figura 5 – Arquitectura de la Plataforma – Elaboración propia

Usuarios

El sistema cuenta con dos grupos de usuarios:

Los usuarios que usan el portal *web* o médicos, conformados por personal de la empresa que deben estar registrados y habilitados dentro del directorio activo de la institución para tener acceso a la plataforma. Desde ella podrán visualizar los pacientes que se encuentren en lista de espera por ser atendidos, realizar videoconferencias a los pacientes de la lista, atender videoconferencias entrantes, visualizar las conversaciones establecidas con los pacientes, enviar mensajes al existir una comunicación abierta, transferencia de archivos PDF e imágenes, y cerrar casos de *chats* atendidos.

Los usuarios que usan el aplicativo móvil o pacientes, deben estar registrados previamente en la base de datos provista por las compañías de seguro que tengan alianza con la empresa, los pacientes deben tener una cuenta para poder acceder al aplicativo en el que podrán realizar videoconferencias mientras se encuentre algún médico disponible, crear nuevas conversaciones de *chat*, enviar archivos y cambiar la contraseña o correo de contacto de la cuenta.

IV.4 Planificación

Mediante el uso del marco de trabajo Scrum se realizó una reunión en donde estuvo presente el equipo del proyecto, conformado por el arquitecto de software Alexander Díaz, desempeñando el rol de Scrum master, Eilyn Millán como gerente de proyecto y Marco Franceschi como desarrollador. En la misma se definió el *Product Backlog* en base al levantamiento de historias de usuario realizado en la etapa de diseño. Una vez definidas todas las actividades se procedió a estimar la prioridad de cada una de ellas y se definieron cuatro *sprints* requeridos para llevar a cabo las actividades que pueden observarse en el *Sprint Backlog* ubicado en el Apéndice A.

IV.5 Sprints

El siguiente apartado describe las actividades realizadas en cada uno de los *sprints* definidos en la etapa de planificación.

IV.5.1 Sprint 1.

Durante el primer sprint se desarrolló el módulo de gestión de usuarios, el cual comprendía aspectos como: Inicio de sesión, Registro de usuarios, Recuperación de contraseña y Modificación de correo o contraseña. Adicionalmente se realizaron actividades como diseño e implementación del modelo de datos y construcción del motor de extracción de datos.

IV.5.1.1 Modelo de datos

Dado la arquitectura del sistema, se estableció un modelo de datos relacional teniendo en cuenta la posible expansión del sistema. Para la representación de la base de datos se utilizó un modelo entidad-relación, en donde se reflejan las entidades y relaciones necesarias para el sistema. La Figura 6 expone el modelo entidad-relación de la plataforma, los nombres de las tablas y atributos fueron modificados por motivos de confidencialidad con los clientes de Trascend.

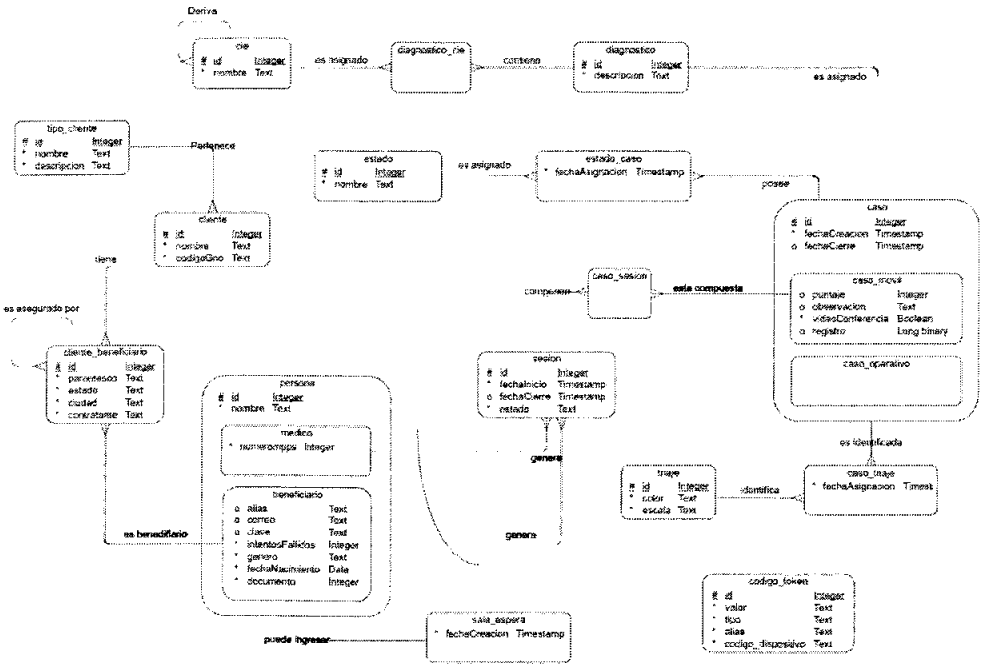

Figura 6 – Modelo entidad-relación – Elaboración propia.

Dada la existencia de un nodo móvil, se diseña e implementa un modelo de datos simplificado para el aplicativo móvil, con el fin de mantener persistencia en el dispositivo a pesar de no tener conexión. Para la creación de la base de datos se utilizó SQLite[41], debido a que es permitido en plataformas con Android y iOS.

[41] **SQLite:** Sistema para gestión de base de datos relacionales contenida en una librería relativamente ligera implementada en C.

IV.5.1.2 Módulo de registro de usuario para aplicativo móvil

Módulo encargado de la creación de nuevas cuentas de usuario para la utilización del aplicativo móvil.

Registro de usuarios

Proceso requerido en caso de que el paciente no posea una cuenta en el sistema, la cual es necesaria para acceder y disfrutar de las funcionalidades de la aplicación móvil. El usuario deberá completar los campos requeridos correctamente, del formulario en la pantalla de registro que se observa en la Figura 7 del Apéndice B, para posteriormente aceptar los términos y condiciones como criterio de creación de una nueva cuenta. El documento de identificación del usuario deberá coincidir con la información suministrada por las compañías de seguro almacenada en la base de datos.

IV.5.1.3 Módulo de autenticación de usuarios del aplicativo móvil:

Es el módulo encargado de gestionar el manejo de los usuarios del aplicativo móvil o pacientes, desde el *FrontOffice*. Se compone de los siguientes sub módulos:

- ***Inicio de sesión***

  Permite el acceso a las funcionales del aplicativo móvil, para poder ingresar se deben llenar todos los datos del formulario mostrado en la Figura 8 del Apéndice B, la autenticación es realizada mediante un correo electrónico y una contraseña previamente ingresada, en caso de que el usuario realice tres intentos fallidos la cuenta será bloqueada. Es necesario tener una cuenta registrada previamente y que no se encuentre bloqueada para poder iniciar sesión. Se utiliza el protocolo abierto OAuth2, el cual provee un mecanismo seguro de autenticación para aplicaciones web, móviles y de escritorio.

- ***Recuperación de contraseña***

  En el caso de no recordar la clave o bloqueo de la cuenta, se deberá acceder a la pantalla para recuperar contraseña reflejada en la Figura 9 del Apéndice B, en la cual se deberá proporcionar el correo electrónico del usuario para realizar un desbloqueo de la cuenta y un reinicio de la clave. Será enviado un correo electrónico a la dirección suministrada previamente con la nueva contraseña generada aleatoriamente para poder ingresar al *FrontOffice*.

- ***Modificación de correo o contraseña***

  Una vez iniciado sesión en el aplicativo móvil es posible consultar los datos de la cuenta a través de una de las opciones mostradas en el menú como se ve en la Figura 10 del Apéndice B, la cual dirige a una pantalla con la información personal del usuario, como se ve en la Figura 11 del Apéndice B, permitiendo la edición del correo electrónico del usuario o contraseña mientras cumpla con las validaciones de los campos pertinentes.

IV.5.1.3 Motor de extracción de datos

El motor es el encargado del procesamiento de los archivos suministrados por las compañías de seguro para agregar o actualizar la información de titulares y beneficiarios. El mismo se desglosa de la siguiente manera:

- **Sub-módulo para la extracción de información de usuarios:** Encargado de todo el proceso desde que detecta un nuevo archivo, su verificación y procesamiento.
- **Sub-módulo para la importación en la base de datos centralizada:** Es el responsable de persistir en la base de datos del sistema toda la información procesada en la extracción de información.

En el siguiente apartado se expone el desarrollo y comportamiento del motor de extracción de datos:

Se creó una serie de directorios en un servidor de pruebas para simular el ambiente en donde los archivos enviados por las aseguradoras serán recibidos a través de una conexión SFTP. Una vez recibido, el motor consulta si se está procesando algún archivo pudiendo leer y procesar hasta 10 archivos en paralelo mediante el uso de hilos, en caso de que exceder la restricción establecida la aplicación encola el archivo mediante RabbitMQ. Una vez que el motor se encuentra disponible, procede a desencolar el archivo para verificar su estructura y en caso de ser correcta, se enviará a un directorio para su procesamiento, si es detectado algún error es indicado en un archivo de *log*[42] y colocará el archivo en un directorio de archivos no procesados.

El formato único del archivo, tendrá una cabecera y un contenido, al mismo se le realizará un análisis sintáctico y posterior a éste, un análisis semántico. El análisis sintáctico consistirá en validar si la cabecera del archivo está formada correctamente y si cada una de sus líneas tiene el contenido adecuado, mientras que el análisis semántico se encarga de validar el contenido de la información suministrada de acuerdo a las reglas de negocio establecidas.

Cuando el archivo sea procesado sin errores, se procederá a almacenar la información extraída en la base de datos continuando a las operaciones de inserción en caso de nuevos registros y actualización para los registros ya existentes.

Debido a las reuniones con el cliente se planteó la posibilidad de la incorporación de distintos formatos de archivos suministrados por distintas empresas aseguradoras, este factor fue tomado en cuenta al diseñar e implementar el motor de extracción de datos, permitiendo la incorporación de nuevos *plugins*

[42] **Log:** registro que almacena las operaciones o eventos ocurridos en un sistema.

(suministrados a través de archivos jar) que permitan el procesamiento de archivos con distintos formatos y validaciones.

El motor se mantendrá en ejecución como un servicio dentro del servidor, monitoreando constantemente los cambios en el directorio de recepción de archivos.

IV.5.2 Sprint 2.

Durante el segundo *sprint* se desarrolló el módulo de autenticación mediante directorio activo, módulo para gestión de lista de espera y el módulo dedicado al manejo de las videoconferencias en el sistema.

IV.5.2.1 Módulo de autenticación de usuarios del portal *web*

Es el módulo que trabaja como enlace entre el directorio activo de la empresa y el BackOffice del sistema. Permite la autenticación de los usuarios del portal *web* a través de su nombre de usuario y contraseña, como puede observarse en la Figura 12 del Apéndice B, mientras que el médico se encuentre registrado y activo en la institución.

IV.5.2.2 Módulo para gestión de lista de espera

Permite la eliminación y agregación de los pacientes a la lista de espera, la cual funciona como una cola, en dónde los usuarios más antiguos serán los primeros en ser atendidos, se describen a continuación las funcionalidades del módulo:

- **Agregar a lista de espera.**

  Los usuarios con acceso al aplicativo móvil pueden seleccionar la opción de registrarse en una lista de espera al ingresar a la pantalla de

videoconferencia mostrada en la Figura 13 del Apéndice B, en caso de no haber médicos disponibles en ese momento que atiendan el caso.

- **Eliminar de la lista de espera.**

  Los usuarios del aplicativo móvil o médicos son los únicos que pueden remover a un paciente de la lista de espera, al contactarlo a través de una videoconferencia. Una vez el doctor se encuentre disponible, podrá consultar quién es el paciente con mayor antigüedad en la lista de espera como muestra la vista reflejada en la Figura 14 del Apéndice B. Al finalizar la sesión del paciente es eliminado de la lista de espera.

IV.5.2.3 Módulo de videoconferencia

Es el módulo encargado de crear casos de videoconferencia y permitir la comunicación entre el paciente y el médico en tiempo real. Esta comunicación puede ser iniciada desde ambos orígenes. A continuación, se describe el proceso de videoconferencia desde sus distintas vías:

BackOffice

Los médicos pueden iniciar videoconferencias siempre y cuando existan pacientes registrados en la lista de espera. Para generar un caso de videoconferencia el usuario del portal *web* debe tener una sesión abierta, en la vista de listado de la página de inicio mostrada en la Figura 14 del Apéndice B, si existe algún paciente esperando por ser atendido podrá observarse una ficha con su nombre y al ser presionada aparecerá una ventana emergente, permitiendo ponerse en contacto con el paciente o cancelar la comunicación. Una vez que un nuevo caso de videoconferencia es generado, se le asigna el estado de "Esperando".

Mientras un paciente de la lista de espera tenga una sesión activa y el aplicativo móvil abierto, puede recibir una notificación en pantalla que indica que un

médico está intentando ponerse en contacto a través de una videoconferencia. Si el paciente decide atender será eliminado de la lista de espera donde posteriormente se agregará el estado "Abierto" al caso generado e iniciará la videoconferencia con el médico que está llamando, tal y como puede verse en la Figura 15 del Apéndice B.

Una vez que el caso haya sido atendido, tanto el médico como el paciente podrán finalizar la comunicación seleccionando la opción correspondiente, lo cual le agrega el estado "Cerrado" y el médico pasa a estar disponible nuevamente.

FrontOffice

Para poder iniciar un nuevo caso de videoconferencia, el paciente o usuario del aplicativo móvil debe haber iniciado sesión y dirigirse a la opción de "Videoconferencia" ubicada en el menú de navegación. Una vez en la pantalla de videoconferencia el usuario podrá seleccionar la opción de "Contactar Médico" presentada en la Figura 13 del Apéndice B, para realizar una video llamada con un médico disponible que es asignado por el sistema dependiendo de su carga de trabajo, concediéndole al caso generado un estado de "Esperando".

Si existen médicos disponibles, el sistema le enviará una notificación a través del portal *web* al médico asignado de que existe una llamada entrante, en caso de aceptar, la llamada se establecerá la comunicación a través de audio y video como puede verse en la Figura 16 del Apéndice B, y se le agregará el estado de "Abierto" al caso de videoconferencia establecido entre el médico y el paciente hasta que alguna de las partes finalice la comunicación, una vez esto ocurra se agrega el estado "Cerrado". Si el médico decide no atender la llamada se cancelará la videoconferencia dándole un estado de "Cerrado" al caso, desocupando al médico asignado.

Cuando no existe ningún médico disponible, el sistema mostrará un mensaje al paciente indicándole que puede anotarse en una lista de espera para ser atendido una vez se desocupen los doctores o en caso de tratarse de una emergencia puede llamar al *Call Center* de la empresa mediante una llamada telefónica a través de la opción "Llamada" ubicada en el menú de navegación. Abriendo la aplicación de marcado del dispositivo con el número telefónico del *Call Center*, reflejada en la Figura 17 del Apéndice B.

IV.5.3 Sprint 3.

Durante el tercer *sprint* se trabajó en el módulo *chat* el cuál maneja todas las comunicaciones escritas entre el médico y el paciente.

IV.5.3.1 Módulo de *chat*

Módulo encargado de crear casos de *chat* y permitir el intercambio de mensajes de texto plano entre el médico y el paciente. Esta comunicación inicia una vez que el paciente o usuario del aplicativo móvil decide iniciar una nueva conversación con el médico a través de su dispositivo.

El paciente debe tener una sesión abierta y seleccionar la opción de "Chats" en el menú de navegación referenciado, esto lo llevará a la pantalla de nuevo chat mostrada en la Figura 18 del Apéndice B, mediante la cual puede generar un nuevo caso de chat con estado "Abierto", con un médico asignado por el sistema dependiendo de su disponibilidad. Al establecer una conexión con el médico, aparecerá la pantalla de *chat* visualizada en la Figura 19 del Apéndice B, mostrando el historial de la conversación con el doctor formado por los mensajes recibidos y los mensajes enviados del paciente, el cuál será almacenado en un archivo con formato XML para su posterior almacenamiento en la base de datos del sistema una vez finalizada la comunicación.

Siempre y cuando el *chat* no haya sido terminado, el paciente podrá consultar los mensajes intercambiados con el médico al ingresar a la conversación. Se crearon restricciones sobre el contenido del mensaje, permitiendo mensajes con un máximo de 300 caracteres y la prohibición de caracteres especiales.

Las comunicaciones establecidas con un médico son listadas en la opción de "Mensajes" en el menú de navegación del portal *web*, desde la vista de listar chats mostrada en la Figura 20 del Apéndice B, donde listan todas las conversaciones en progreso con pacientes a través de una ficha con el nombre del paciente y el número de mensajes no leídos. Al seleccionar el nombre del paciente se puede acceder a la vista de *chat* visualizada en la Figura 21 del Apéndice B, desde la misma se observan los mensajes que son intercambiados con el paciente. Los mensajes enviados desde el portal *web* cuentan con las mismas limitaciones de contenido que en el aplicativo móvil. En la vista de listar *chats* los usuarios del portal *web* tienen la capacidad de culminar un chat en progreso al presionar el icono de *check* en la vista de listar chats, agregando el estado de "Cerrado" al caso seleccionado.

Una vez culminado, el contenido del archivo XML en donde se guarda el historial de la conversación, es enviado al sistema para ser almacenado en la base de datos como registro del caso *chat* para su posterior auditoria.

Si no existe ningún doctor disponible al intentar iniciar una nueva conversación, se le mostrará una ventana emergente al paciente, indicando que debe esperar a que algún médico se desocupe para volver a intentar.

IV.5.4 Sprint 4.

Durante el cuarto *sprint* se trabajó en el módulo de transferencia de archivos el cual permite el envío de datos adjuntos en las conversaciones establecidas entre los médicos y los pacientes.

IV.5.4.1 Módulo de transferencia de archivos

Módulo que permite el envío de archivos PDF e imágenes con extensión JPG[43] o PNG[44], entre los usuarios del aplicativo móvil y los usuarios del portal *web*. Compuesta a través de tres sub-módulos que interactúan entre sí:

- **Sub-módulo para la lectura de archivos:** Realiza todas las operaciones relacionadas con la lectura en disco para la selección del fichero a enviar utilizando la interfaz de programación para manejo de archivos (API *File*) disponible por defecto en navegadores *web*.
- **Sub-módulo para la conversión de archivos**: Se encarga de utilizar los datos obtenidos a través de la lectura de archivos, para luego transformarlos en el formato adecuado y enviarlos a través de la conexión establecida por webRTC.
- **Sub-módulo para la recepción de archivos**: Recibe el contenido del archivo y es responsable de mostrarlo en pantalla.

A continuación, se describe detalladamente el funcionamiento del módulo de transferencia de archivos para el aplicativo móvil y el portal *web*.

Para el envío de archivos es necesario que exista una conversación en progreso, en la pantalla de *chat* se debe presionar el icono del *clip* para que se active el explorador de archivos con el cual se podrá seleccionar el archivo a enviar sin superar el tamaño de 1 MB. Posterior a esto, se mostrará una ventana emergente que permitirá confirmar o cancelar el envío del archivo seleccionado. Cuando el archivo es enviado se podrá ver una miniatura de la imagen transferida en caso de tratarse de un PNG o JPG, mientras que para los archivos PDF se mostrará un ícono indicando que es un documento.

---

[43] **JPG:** Joint Photographic Experts Group, formato de comprensión de imágenes en color como en escala de grises.

[44] **PNG:** Portable Network Graphics, formato de comprensión de imágenes no sujeto a patentes, que no posee pérdida de calidad.

Al presionar sobre alguna de las miniaturas en pantalla desde el dispositivo móvil se abrirá el archivo con la aplicación utilizada por defecto para esa extensión y en caso de no estar configurada permitirá elegir la aplicación con la que se pueda abrir el archivo. Los archivos recibidos por el médico serán almacenados en la memoria del teléfono una vez abiertos, alojándose en un directorio creado al instalarse la aplicación. Mientras que en el portal web las miniaturas en la conversación se abren en una nueva pestaña en el navegador mostrando el archivo enviado o recibido.

El proceso de envío de archivos desde el aplicativo móvil se puede observar en la Figura 22 del Apéndice B, mientras que la Figura 23 del Apéndice B muestra el mismo proceso desde el portal web.

CAPÍTULO V: RESULTADOS

El resultado del desarrollo es una plataforma para la comunicación multimedia en tiempo real compuesta por un aplicativo móvil utilizado por los pacientes para establecer comunicación con los médicos a través de un portal *web*, además de un motor para la extracción de datos con la finalidad de procesar la información suministrada por compañías de seguro, ampliando la gama de servicios de la empresa contratante y brindar una mejor asistencia a sus clientes.

Los médicos o usuarios del portal *web*, atienden consultas, realizan diagnósticos y brindan orientación a los usuarios del aplicativo móvil a través de videoconferencias o comunicaciones escritas. Utilizan la plataforma como un medio dentro de las instalaciones de la empresa para mejorar la calidad de su servicio, logrando atender múltiples casos a la vez y cuando lo amerite, el envío de indicaciones en formato PDF o una imagen.

Los pacientes o usuarios del aplicativo móvil, pueden solicitar atención médica a través de distintos canales, como llamar al *Call Center* de la empresa, iniciar una videoconferencia con algún doctor disponible o establecer chats en dónde pueden enviar archivos PDF o imágenes que respalden sus consultas. También es posible la consulta y modificación de alguno de sus datos.

Al final del ciclo de desarrollo se procedió a la instalación y configuración de la solución en las instalaciones de la empresa, verificando el correcto funcionamiento de cada una de las partes del sistema, y realizando un periodo de prueba por parte del cliente y personal de la empresa.

A continuación, se detalla el resultado obtenido en cada uno de los objetivos planteados para la culminación del sistema.

Diseñar e implementar el modelo de base de datos para el sistema.

Se diseñó e implementó un modelo de datos para el manejo de información, teniendo en cuenta los planes de la empresa para la expansión de sus sistemas, dando como resultado una base de datos centralizada para almacenar la información transaccional de la plataforma y una base de datos simplificada para el aplicativo móvil para persistir los datos de sesión del usuario. Se puede detallar en el capítulo IV, Sprint 1, sección 5.1.1.

Desarrollar una API RESTful para la interoperabilidad de los componentes del sistema.

Se desarrolló una interfaz de programación que permite acceder a la lógica de negocio de cada uno de los módulos implementados para el funcionamiento del sistema. Los servicios son hospedados en uno de los servidores de la empresa, publicando las rutas a distintas funcionalidades como "Realizar videoconferencia" o "Autenticar Paciente", la información es transmitida a través de formato JSON.

Se obtuvo como resultado una capa de comunicación basada en servicios que conecta los módulos interdependientes de la plataforma.

Desarrollar el módulo para la extracción de la información de clientes de empresa aseguradora.

Se desarrolló un módulo para detectar, extraer y procesar información, el mismo trabaja como un servicio en un servidor Linux dentro de las instalaciones de la empresa, manteniéndose a la espera de cualquier cambio sobre el directorio dónde se reciben los archivos de texto planos suministrados a través de una conexión SFTP configurada para garantizar el envío seguro de la información.

Se obtuvo como resultado un motor para la extracción de datos y su procesamiento a través de archivos suministrados por las empresas aseguradoras aliadas con el cliente contratante. Los archivos procesados contienen la información de los titulares y beneficiarios que pueden utilizar el aplicativo móvil.

El proceso del módulo puede ser observado en el capítulo IV, Sprint 1, sección 5.1.3.

Desarrollar módulo de registro de usuario para aplicativo móvil.

Se obtuvo un módulo encargado de la creación de cuentas de usuario para el uso del aplicativo móvil, a través de un formulario que solicita el número de cédula, correo electrónico y contraseña utilizados posteriormente para tener acceso a la aplicación. Un paciente puede crear una cuenta si la cédula proporcionada en el formulario coincide con un titular o beneficiario registrado en la base de datos centralizada. Todo usuario que cree una nueva cuenta deberá estar de acuerdo con los términos y condiciones mostrados en pantalla una vez se procede con el proceso de creación. El detalle del módulo se encuentra en el capítulo IV, Sprint 1, sección 5.1.2.

Desarrollar módulo de autenticación de usuarios.

Se obtuvo un módulo para permitir el acceso a las funcionalidades de la plataforma a sus distintos usuarios. Empleando la autenticación y acceso de los usuarios del aplicativo móvil. También garantiza el acceso al portal *web* mediante una conexión al directorio activo de la empresa. El módulo también comprende la recuperación de contraseña en caso de olvido o bloqueo y la consulta de los datos de la cuenta, así como también la modificación del correo o contraseña. Se puede encontrar el detalle de este módulo en el capítulo IV, el cual está segmentado en el Sprint 1, sección 5.1.3 referente a la autenticación de usuarios del aplicativo móvil y en el Sprint 2, sección 5.2.1 acerca de la autenticación de usuarios del portal web.

Desarrollar módulo de video conferencia.

Se desarrolló un módulo para establecer comunicación de audio y video en tiempo real vía *streaming*, permitiendo establecer conexiones punto a punto entre los pacientes y médicos generando nuevos casos de videoconferencia. El detalle de este módulo se puede muestra en el capítulo IV, Sprint 2, sección 5.2.3.

Desarrollar módulo de *chat*.

Se desarrolló un módulo para el intercambio de mensajes escritos entre un paciente y un médico en tiempo real, contemplando la creación y cierre de nuevos casos y permitiendo mandar mensajes y en el caso de los médicos, mantener múltiples conversaciones con distintos pacientes. Este módulo es detallado en el capítulo IV, Sprint 3, sección 5.3.1.

Desarrollar módulo de transferencia de archivos.

Se desarrolló un módulo que permite la transferencia de documentos PDF o imágenes con los formatos JPG y PNG entre un paciente y un médico, mientras los archivos no superen un peso máximo de 1 MB. El detalle de este módulo se observa en el capítulo IV, Sprint 4, sección 5.4.1.

Rediseñar el proceso de comunicación entre pacientes y doctores a través de un aplicativo móvil.

Se rediseñó el proceso de atención al paciente a través de la creación de un aplicativo móvil y un portal *web* que permiten establecer comunicaciones efectivas en tiempo real:

- El aplicativo móvil conlleva a la apertura de nuevos canales para los pacientes, mediante mensajes escritos (*chats*), vía videoconferencia y transferencia de archivos (.pdf, .png y .jpg).

- Los especialistas de la salud podrán prestar un mejor servicio con el uso de un portal *web*, permitiéndoles mantener el contacto con los pacientes con acceso al aplicativo móvil, al poder atender videoconferencias o chats.

Con la inclusión de las nuevas vías de comunicación a través de las herramientas tecnológicas nombradas previamente, se puede proporcionar la suficiente información para realizar diagnósticos certeros sin la necesidad de concretar una cita. El proceso resultante de este apartado puede ser visualizado en el capítulo IV, sección 1 mediante la Figura 3 – Interacción de la Plataforma.

Evaluar el uso de "*frameworks*" de aplicaciones híbridas para la implementación del aplicativo móvil.

Mediante una investigación y estudio de distintos *frameworks* para el desarrollo de aplicaciones híbridas, se determinó que el más adecuado para cumplir con las necesidades del proyecto es el Ionic. Se evaluaron aspectos claves como lo son:

- Compatibilidad sobre funciones de *hardware* en distintos dispositivos móviles.
- Apariencia semejante al diseño nativo con posibilidad de personalización.
- Comunidad activa que permita el mantenimiento y mejora constante del *framework*.
- Documentación suficiente y de calidad para su entendimiento y aprendizaje.

Evaluar tecnologías disponibles para la comunicación en tiempo real a través de conexiones punto a punto.

Mediante una investigación y estudio de distintos protocolos y tecnologías para la comunicación en tiempo real a través de conexiones punto a punto, se determinó que el protocolo más adecuado para cumplir con las exigencias del proyecto es el RTP/RTCP, a partir de allí se concluyó que la tecnología más apropiada para el desarrollo del sistema es WebRTC. Se evaluaron aspectos importantes para su selección como lo son:

- Estabilidad y confianza de la transmisión.
- Herramientas de desarrollo disponibles en el mercado.
- Seguridad de los datos transmitidos.
- Compatibilidad y soporte de navegadores.

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentan las palabras finales sobre este Trabajo Instrumental de Grado, presentando las conclusiones del proyecto y sus recomendaciones.

VI.1 Conclusiones

Por medio de este trabajo de grado se diseñó e implementó una plataforma para la comunicación multimedia en tiempo real como producto de *software*, la cual a través de sus distintos componentes brinda un aporte funcional al sector de la salud, creando nuevos canales de comunicación entre pacientes y médicos.

Durante el análisis, diseño, planificación y desarrollo del proyecto se constataron varios aprendizajes, entre los cuales vale la pena resaltar:

- WebRTC como próximo estándar para la comunicación punto a punto para la *web*, brindando una buena calidad y estabilidad para transmisiones de audio, video y datos. Gran versatilidad por no necesitar *plugins* para conectar dos o más clientes.

- Importancia y facilidad de uso de proveedores de *Platform as a Service* como lo es Xirxys o Twilio que a través de suscripciones es posible contar con toda la infraestructura necesaria para encargar de un número de operaciones computacionales sin necesidad de tener un servidor instalado y configurado.

- Importancia del desarrollo híbrido, en especial en el mercado de los aplicativos móviles, en donde a través de una misma base de código es posible generar aplicaciones funcionales y con un buen rendimiento para distintas plataformas.

Los pacientes que sean titulares o beneficiarios de una póliza de seguro de alguna de las empresas aseguradoras, tienen acceso a un aplicativo móvil con el que pueden ponerse en contacto con el personal médico de la empresa, a través de videoconferencias, conversaciones escritas o chats y transferir documentos en formato PDF o imágenes. Los médicos podrán atender las solicitudes de los clientes por medio de un portal *web* que se mantendrá operativo dentro de las instalaciones del cliente.

VI.2 Recomendaciones

Durante el desarrollo de este Trabajo Instrumental de Grado (T.I.G.) se evidenciaron ciertos puntos de mejora o la inclusión de nuevas funcionalidades que permitan mejorar la experiencia de usuario y aumentar la mantenibilidad del sistema:

- Incorporación del lenguaje de programación TypeScript[45] para el desarrollo del portal *web*, en sustitución de JavaScript que fue seleccionado por Trascend debido a poseer mayor tiempo en el mercado. Esto con la finalidad de mejorar la mantenibilidad y refactorización del código, al permitir estructuras complejas como interfaces, datos tipados, entre otros.

- Actualización de la versión del aplicativo Ioni cv1 a Ionic v2, esto con el fin de incorporar mejoras en cuanto a rendimiento y compatibilidad con distintos dispositivos.

- Adaptación de interfaz gráfica según la identidad corporativa del cliente, tanto para los usuarios del aplicativo móvil como para los usuarios del portal *web*.

- Futura incorporación de llamadas al *Call Center* a través de VoIP, usando las capacidades provistas por webRTC como tecnología.

- Selección de múltiples documentos o imágenes para la transferencia de archivos por lotes desde el paciente o el médico.

- Implementar un módulo de notificaciones para el aplicativo móvil, que indique cuándo un médico intente establecer comunicación con el paciente.

[45] **TypeScript:** Es un lenguaje que provee un conjunto de características adicionales (clases, interfaces, enumeraciones, entre otros) que luego es compilado a JavaScript.

BIBLIOGRAFÍA

[1] JavaScript. [En línea]. Disponible: https://developer.mozilla.org/es/docs/Web/JavaScript *[Último acceso:* 7 de mayo de 2017]

[2] *Acerca de JavaScript.* [En línea]. Disponible: https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript [Último acceso: 7 de mayo de 2017]

[3] *HTML.* [En línea]. Disponible: https://developer.mozilla.org/es/docs/Web/HTML [Último acceso: 8 de mayo de 2017]

[4] *Guía Breve de CSS.* [En línea]. Disponible: http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo [Último acceso: 8 de mayo de 2017]

[5] *CSS.* [En línea]. Disponible: https://developer.mozilla.org/es/docs/Web/CSS [Último acceso: 8 de mayo de 2017]

[6] *Protocolo UDP.* [En línea]. Disponible: http://es.ccm.net/contents/284-protocolo-udp [Último acceso: 10 de mayo de 2017]

[7] *Diferencias entre los protocolos TCP y UDP.* [En línea]. Disponible: http://es.ccm.net/faq/1559-diferencias-entre-los-protocolos-tcp-y-udp [Último acceso: 10 de mayo de 2017]

[8] *¿App nativa, web o híbrida?.* [En línea]. Disponible: http://www.raona.com/es/Solutions/Template/163/App-nativa-web-o-h%C3%ADbrida- [Último acceso: 11 de mayo de 2017]

[9] *Introducción a SOA y servicios web.* [En línea]. Disponible: https://www.ibm.com/developerworks/ssa/webservices/newto/service.html [Último acceso: 12 de mayo de 2017]

[10] *Guía Breve de Servicios Web.* [En línea]. Disponible: http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb [Último acceso: 12 de mayo de 2017]

[11] *HTTP - Hypertext Transfer Protocol*. [En línea]. Disponible: https://www.w3.org/Protocols/ [Último acceso: 12 de mayo de 2017]

[12] *JPA (Java Persistence API)*. [En línea]. Disponible: https://www.ibm.com/support/knowledgecenter/es/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/cwlp_jpa.html [Último acceso: 13 de mayo de 2017]

[13] *The Java EE 6 Tutorial*. [En línea]. Disponible: http://docs.oracle.com/javaee/6/tutorial/doc/bnbpz.html [Último acceso: 13 de mayo de 2017]

[14] *Ejemplo de JPA , Introducción (I)*. [En línea]. Disponible: http://www.arquitecturajava.com/ejemplo-de-jpa/ [Último acceso: 13 de mayo de 2017]

[15] *RTP*. [En línea]. Disponible: https://www.voip-info.org/wiki/view/RTP [Último acceso: 14 de mayo de 2017]

[16] *About HTML5 WebSocket*. [En línea]. Disponible: https://www.websocket.org/aboutwebsocket.html [Último acceso: 14 de mayo de 2017]

[17] *The WebSocket Protocol*. [En línea]. Disponible: https://tools.ietf.org/html/rfc6455 [Último acceso: 14 de mayo de 2017]

[18] *The Secure Real-time Transport Protocol (SRTP)*. [En línea]. Disponible: https://tools.ietf.org/html/rfc3711 [Último acceso: 14 de mayo de 2017]

[19] *Real-Time Streaming Protocol Version 2.0*. [En línea]. Disponible: https://tools.ietf.org/html/rfc7826 [Último acceso: 15 de mayo de 2017]

[20] *Getting Started with WebRTC*. [En línea]. Disponible: https://www.html5rocks.com/en/tutorials/webrtc/basics/ [Último acceso: 16 de mayo de 2017]

[21] *WebRTC API*. [En línea]. Disponible: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API [Último acceso: 16 de mayo de 2017]

[22] *Qué es Streaming*. [En línea]. Disponible: https://desarrolloweb.com/articulos/482.php [Último acceso: 16 de mayo de 2017]

[23] *What Is a Firewall?*. [En línea]. Disponible: http://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html [Último acceso: 20 de mayo de 2017]

[24] *HTML5*. [En línea]. Disponible: https://developer.mozilla.org/es/docs/HTML/HTML5 [Último acceso: 20 de mayo de 2017]

[25] *¿Qué es un datagrama?*. [En línea]. Disponible: http://www.viu.es/que-es-un-datagrama/ [Último acceso: 20 de mayo de 2017]

[26] *What is Fallback and What Does it Mean?*. [En línea]. Disponible: https://blogs.technet.microsoft.com/cmpfekevin/2013/03/05/what-is-fallback-and-what-does-it-mean/ [Último acceso: 21 de mayo de 2017]

[27] *Java EE at a Glance*. [En línea]. Disponible: http://www.oracle.com/technetwork/java/javaee/overview/index.html [Último acceso: 7 de mayo de 2017]

[28] *Building RESTful Web Services with JAX-RS*. [En línea]. Disponible: http://docs.oracle.com/javaee/6/tutorial/doc/giepu.html [Último acceso: 23 de mayo de 2017]

[29] *Oracle MySQL*. [En línea]. Disponible: https://www.oracle.com/lad/mysql/index.html [Último acceso: 23 de mayo de 2017]

APÉNDICES

Apéndice A – Artefactos de la metodología

Tabla 3 – Historia de usuario 001.

Historia de usuario - GNO-WRTC					
Número	001	**Creador**	Elyn Millán	**Actor**	Sistema
Nombre	Extraer datos asegurados				
Descripción	Los datos de los asegurados se deben extraer de un archivo txt que provee la aseguradora.				
Observaciones	-				
Criterios de Aceptación	1. El archivo con el listado de asegurados debe ser de tipo txt. 2. Los archivos serán cargados por la aseguradora a través de una tarea automatizada de conexión remota a un Servidor SFTP. 3. El proceso que carga el archivo en la ubicación para extracción se ejecutará diariamente a partir de las 07:30 p.m. con un aproximado de procesamiento de 1 hora 30 minutos hasta 2 horas. 4. GNO deberá mover el archivo hasta un directorio de procesamiento. 5. Cada archivo se leerá línea a línea y se contará secuencialmente cada registro (uno por uno). 6. Diariamente se deberá validar la información enviada para identificar: 7. Nuevos asegurados e insertarlos. 8. Actualizaciones sobre asegurados existentes. 9. Eliminaciones de asegurados, es decir, que no vienen en el archivo pero existen en base de datos. 10. Se deberá hacer un análisis sintáctico del formato del archivo, que cumpla con el formato indicado. 11. Los registros que no cumplen con el formato serán identificados y almacenados en un archivo que se puede configurar para reprocesar. 12. Se deberá hacer un análisis semántico de los campos encontrados por cada registro: 13. Se generará un log resumen o general con el nombre del archivo, fecha, hora y un estado de procesado o no procesado. 14. Se generará un log de detalle, con el detalle del archivo no procesado, se debe mostrar el identificador del registro, el número de línea, la fecha y la hora, así por cada registro que no cumpla. 15. Si un archivo se procesa con éxito el archivo se debe mover a un directorio de "Archivos Procesados". 16. Si el proceso de extracción falla por un error no controlado se deberá dejar registro en un log, con el nombre del archivo, indicando el error que origina la falla, la fecha y la hora. 17. El archivo que no se procese por fallas, debe moverse a un directorio de "Archivos no Procesados". 18. El archivo deberá tener el siguiente formato:				

Primer Registro (Cabecera):

Nombre del campo	Tipo	Longitud Máxima	Observaciones:
Cliente	Alfanumérico	12	Ingrese el Código Asignado por Grupo Nueve Once de la empresa (GNO), incluyendo el primer carácter alfabético según el tipo de cliente; el acrónimo se construye con las primeras letras de su razón social ejemplo Estar Seguros (ES), seguido del número de registro asignado a la empresa. Ejemplo: GNOES040
Nombre Contratante	Alfabético	40	Ingrese el nombre y apellido del contratante. Ejemplo: CIGARRERA BIGOTT
Nombre Titular	Alfabético	40	Ingrese el nombre y apellido del titular. Ejemplo: ALEJANDRA MARIA DIAZ VELIZ
N° C.I Titular	Numérico	15	Ingrese el número de documento del titular de la póliza. Ejemplo: 15700581
Carga Familiar	Numérico	2	Ingrese el total de cargas familiares que posee el titular de la póliza. Ejemplo: 10
Parentesco con el Titular	Alfabético	40	Este campo referencial tiene como fin facilitar a la empresa el parentesco con el titular de la póliza con carácter de control interno, por lo que podrá ingresar la referencia o dejar vacío el campo. Recuerde que las referencias no deben repetirse, y debes llevar una secuencia (CAMPO OPCIONAL), Ejemplo PADRE

Cuerpo del Archivo (Detalle):

Cédula del Familiar	Numérico	15	Ingrese el número de documento del empleado. Ejemplo: 15700581
Fecha de Nacimiento	Numérico	10	Ingrese la fecha de Nacimiento, titular, beneficiario del titular, siguiendo el formato (dd/mm/yyyy). Ejemplo: 01/07/2013.
Edad	Numérico	2	Ingrese la Edad del Titular, beneficiario del titular. Ejemplo: 27
Sexo	Alfabético	10	Ingrese el sexo del titular, beneficiario del titular. Ejemplo: MASCULINO
Ramo	Alfabético	40	Ingrese el ramo al que posee contratado, activo, bien sea colectivo o particular, titular, beneficiario del titular. Ejemplo AP, AP ESCOLAR
Estado	Alfabético	40	Ingrese estado donde se encuentra contratado, activo, bien sea colectivo o particular, titular, beneficiario del titular. Ejemplo CARABOBO
Ciudad	Alfabético	40	Ingrese la ciudad donde se encuentra contratado, activo, bien sea colectivo o particular, titular, beneficiario del titular. Ejemplo VALENCIA

Tabla 4 – Historia de usuario 002.

Historia de usuario - GNO-WRTC					
Número	002	**Creador**	Eilyn Millán	**Actor**	Paciente
Nombre	Registrar Usuario				
Descripción	Como asegurado quiero poder registrar un usuario y la clave para hacer uso de la aplicación móvil.				
Observaciones	-				
Criterios de Aceptación	1. Se deberá ingresar el número de cédula del usuario para validar si pertenece alguna aseguradora que posea el servicio. 2. Se podrá registrar una clave para autenticarse en el aplicativo, la clave debe cumplir con lo siguiente: a. Debe contener al menos 6 caracteres. b. Debe tener máximo 30 caracteres. c. Se podrán agregar caracteres especiales del tipo: "_","-". 3. Se deberá confirmar la clave y esta confirmación debe coincidir con lo registrado anteriormente. 4. Se podrá registrar un correo electrónico valido.				

Tabla 5 – Historia de usuario 003.

Historia de usuario - GNO-WRTC					
Número	003	**Creador**	Eilyn Millán	**Actor**	Paciente
Nombre	Autenticar Usuario				
Descripción	Como asegurado quiero poder hacer login en el aplicativo móvil con mi correo y mi clave.				
Observaciones	-				
Criterios de Aceptación	1. El asegurado se podrá autenticar en el aplicativo móvil colocando un correo y una clave. 2. El asegurado se podrá autenticar en el aplicativo móvil colocando en el campo usuario el correo electrónico y será válido para el ingreso. 3. El ingreso incorrecto de usuario y/o clave de forma consecutiva en 3 intentos consecutivos bloqueará el usuario. 4. El asegurado podrá desbloquear su usuario o clave utilizando la opción recuperar contraseña.				

Tabla 6 – Historia de usuario 004.

Historia de usuario - GNO-WRTC					
Número	004	**Creador**	Eilyn Millán	**Actor**	Paciente
Nombre	Recuperar Clave				
Descripción	Como asegurado deseo poder recuperar clave por olvido o bloqueo.				
Observaciones	-				
Criterios de Aceptación	1. Se podrá recuperar la clave ingresando un correo electrónico. 2. Se podrá desbloquear el usuario ingresando un correo electrónico válido. 3. Se le enviará una clave temporal al correo registrado. 4. El asegurado podrá modificar la clave temporal o el correo electrónico a través del menú principal una vez haya iniciado sesión.				

Tabla 7 – Historia de usuario 005.

Historia de usuario - GNO-WRTC					
Número	005	**Creador**	Eilyn Millán	**Actor**	Paciente
Nombre	Modificar Clave y Correo Electrónico				
Descripción	Como asegurado deseo poder modificar la clave temporal que se ha enviado a mi correo, o modificar el correo que tengo registrado.				
Observaciones	-				
Criterios de Aceptación	1. Se podrá modificar la clave tanto la temporal que se ha enviado al correo del usuario para recuperación como la clave actual, introduciendo la clave temporal o actual y la nueva clave. 2. El asegurado podrá modificar el correo electrónico que tiene agregado en la cuenta.				

Tabla 8 – Historia de usuario 006.

Historia de usuario - GNO-WRTC					
Número	006	**Creador**	Eilyn Millán	**Actor**	Paciente
Nombre	Cerrar Sesión				
Descripción	Como asegurado deseo poder salir del aplicativo móvil cerrando la sesión.				
Observaciones	-				
Criterios de Aceptación	1. Se podrá cerrar la sesión desde una opción del menú principal del aplicativo. 2. Cerrar la se sesión enviará al usuario a la página de inicio para hacer login.				

Tabla 9 – Historia de usuario 007.

Historia de usuario - GNO-WRTC					
Número	007	**Creador**	Eilyn Millán	**Actor**	Paciente
Nombre	Gestionar Lista de Espera				
Descripción	Como asegurado quiero tener la opción de ingresar a una lista de espera hasta que se desocupe un médico y me atienda por videoconferencia.				
Observaciones	-				
Criterios de Aceptación	1. Se tendrá una opción en pantalla para ingresar a una lista de espera para ser atendido cuando el médico esté disponible. 2. Al ingresar a la opción de lista de espera el asegurado deberá esperar un tiempo prudente para que el médico que se desocupe primero le devuelva la llamada por videoconferencia. 3. Se gestionará una única lista de espera y cada asegurado será atendido en el orden en el que ingresaron a la lista.				

Tabla 10 – Historia de usuario 008.

Historia de usuario - GNO-WRTC					
Número	008	**Creador**	Eilyn Millán	**Actor**	Paciente
Nombre	Realizar Video Conferencia				
Descripción	Como asegurado quiero poder hacer videoconferencia con un médico para que me de orientación médica y un diagnóstico a mi caso.				
Observaciones	-				
Criterios de Aceptación	1. Se podrá iniciar una videoconferencia con un médico disponible en el momento de iniciar la comunicación. a. El asegurado no podrá seleccionar el médico con el que desee hablar. b. Al iniciar la videoconferencia se asignará un médico automáticamente si hay alguno disponible. 2. Si no hay disponibilidad de atención médica al momento de la llamada. a. Se mostrará un mensaje en pantalla notificando que debe esperar para ser atendido y se sugerirá llamar al call center en caso de ser emergencia. b. El asegurado tendrá la opción de presionar un botón de lista de espera para que se le devuelva la llamada. 3. Validará - La videoconferencia terminará cuando cualquiera de los extremos de la comunicación decida concluir la videoconferencia.				

Tabla 11 – Historia de usuario 009.

Historia de usuario - GNO-WRTC					
Número	009	**Creador**	Eilyn Millán	**Actor**	Paciente
Nombre	Realizar Chat				
Descripción	Como asegurado quisiera poder escribir por chat a un médico y comentar el malestar o síntomas que me aquejan.				
Observaciones	-				
Criterios de Aceptación	1. Se podrá hacer una conversación con los médicos que estén prestando el servicio en un momento determinado mediante el chat. a. El médico no se podrá seleccionar b. Se evaluará el límete de chats abiertos por cada médico para asignar el nuevo chat 2. Una conversación por chat se podrá dar por terminada cuando el asegurado cierre la ventana de chat. 3. El médico podrá dar por terminada la conversación sino recibe respuesta del paciente. 4. Una vez establecida una comunicación no se podrá iniciar una nueva con otro médico que esté disponible. 5. El podrá asegurado podrá enviar pdf, jpg y png a través del chat. 6. Las imágenes podrán tener un máximo de 1MB 7. Se enviará un mensaje en caso que sobrepase ese límite para que modifique el archivo y se vuelva a cargar.				

Tabla 12 – Historia de usuario 010.

Historia de usuario - GNO-WRTC					
Número	010	**Creador**	Eilyn Millán	**Actor**	Paciente
Nombre	Llamar Call Center				
Descripción	Como asegurado quiero poder contactar el call center desde la aplicación móvil de Grupo Nueve Once.				
Observaciones	_				
Criterios de Aceptación	1. Se tendrá una opción en pantalla para hacer una llamada al call center de Grupo Nueve Once.				

Tabla 13 – Historia de usuario 011.

Historia de usuario - GNO-WRTC					
Número	011	**Creador**	Eilyn Millán	**Actor**	Médico
Nombre	Autenticar Médico				
Descripción	Como médico quiero poder autenticarme en el aplicativo web para atender a los pacientes.				
Observaciones	_				
Criterios de Aceptación	1. El médico se podrá autenticar en el aplicativo web colocando su cédula, el número de su MPPS y la clave que tiene en el Directorio activo de Grupo de Nueve Once. 2. En caso de ingresar clave errar favor comunicarse con el departamento de TI.				

Tabla 14 – Historia de usuario 012.

Historia de usuario - GNO-WRTC					
Número	012	**Creador**	Eilyn Millán	**Actor**	Médico
Nombre	Atender Video Conferencia				
Descripción	Como médico quisiera poder atender la videoconferencia con las consulta médicas de los pacientes.				
Observaciones	_				
Criterios de Aceptación	1. Se desplegará en pantalla la videoconferencia que podrá atender. 2. Se podrá atender una videoconferencia a la vez. 3. Se puede dar por terminada una videoconferencia si no se recibe comunicación de parte del paciente o asegurado. 4. Se podrá dar un diagnóstico de la videoconferencia realizada con el paciente o asegurado.				

Tabla 15 – Historia de usuario 013.

Historia de usuario - GNO-WRTC					
Número	013	**Creador**	Eilyn Millán	**Actor**	Médico
Nombre	Atender Chat				
Descripción	como médico quiero poder atender a mis pacientes mediante un chat.				
Observaciones	_				
Criterios de Aceptación	1. Se podrán tener hasta diez (10 número configurable) chats activos para atender. 2. Se podrán recibir las imágenes que envíe el paciente a través del chat. 3. El médico podrá enviar un pdf con el récipe de las medicinas de ser necesario o alguna imagen (de tipo jpg y png) si así lo requiere. 4. El archivo o imagen que se envíe no podrá pesar más de 1MB. 5. El médico podrá cerrar la sesión de chat cuando el paciente no responda en un tiempo determinado. 6. Se almacenará histórico de las conversaciones que entabla el médico para revisiones posteriores en base de datos.				

Tabla 16 – Historia de usuario 014.

Historia de usuario - GNO-WRTC					
Número	014	**Creador**	Eilyn Millán	**Actor**	Médico
Nombre	Atender Pacientes en Sala de Espera				
Descripción	como médico quiero poder atender pacientes en Sala de espera para videoconferencia.				
Observaciones	_				
Criterios de Aceptación	1. El médico podrá atender pacientes de la lista única de asegurados en espera. 2. Deberá seleccionar atender sala de espera, lo que iniciará una videoconferencia con el paciente que se encuentre de primero en esta lista de espera. 3. Se dará atención a la sala de espera en el orden en el que ingresaron cada vez que un médico se desocupe y presione la opción atender sala de espera.				

Tabla 17 – Sprint backlog 1.

Código Actividad	Actividad	Estado	Iteracion (Sprint)	Historia de usuario (HU)
1-000-1	Diseñar e implementar modelo de base de datos centralizada.	Terminado.	1	-
1-000-2	Diseñar e implementar modelo de base de datos para el dispositivo móvil.	Terminado.	1	-
1-000-3	Generar mapeo de entidades para el manejo del ORM (JPA).	Terminado.	1	-
1-001-4	Crear coordinador de hilos para el manejo de múltiples archivos del motor que extrae información de las aseguradoras.	Terminado.	1	001
1-001-5	Implementar reglas de negocio necesarias para el procesamiento de la aseguradora.	Terminado.	1	001
1-001-6	Desarrollo de reporte de errores y logs del motor de extracción de datos.	Terminado.	1	001
1-001-7	Implementar reglas de negocio necesarias para el procesamiento de la aseguradora.	Terminado.	1	001
1-001-8	Implementación de persistencia de datos del archivo en la base de datos centralizada.	Terminado.	1	001
1-000-9	Creación de proyecto de ionic para el aplicativo móvil.	Terminado.	1	-
1-003-10	Diseñar e implementar el formulario de inicio de sesión para el aplicativo móvil.	Terminado.	1	003
1-002-11	Diseñar e implementar el formulario de registro para el aplicativo móvil.	Terminado.	1	002
1-004-12	Diseñar e implementar el formulario de recuperación de contraseña o usuario bloqueado para el aplicativo móvil.	Terminado.	1	004
1-005-13	Diseñar e implementar el módulo para la consulta/modificación de la cuenta del usuario.	Terminado.	1	005
1-005-14	Implementación de métodos para el manejo de la Base de datos	Terminado.	1	-
1-004-15	Diseño e implementación del servicio y lógica del proceso recuperación de contraseña.	Terminado.	1	004
1-005-16	Diseño e implementación del servicio y lógica del proceso consulta/modificación de la cuenta del usuario.	Terminado.	1	005
1-002-17	Diseño e implementación del servicio y lógica del proceso de registro de nuevos usuarios.	Terminado.	1	002
1-010-18	Implementación de realizar llamada telefónica mediante el aplicativo móvil.	Terminado.	1	010
1-006-19	Implementación de cierre de sesión desde el aplicativo móvil.	Terminado.	1	006

Tabla 18 – Sprint backlog 2.

Código Actividad	Actividad	Estado	Iteracion (Sprint)	Historia de usuario (HU)
2-007-1	Diseño e implementación del servicio y lógica del manejo de emergencias pendientes o sala de espera.	Terminado.	2	007
2-007-2	Diseño e implementación del servicio y lógica del manejo de videoconferencias para el beneficiario.	Terminado.	2	007
2-008-3	Diseño e implementación del servicio y lógica del manejo de videoconferencias para el médico.	Terminado.	2	008
2-008-4	Diseñar e implementar pantalla para el módulo de videoconferencia en el aplicativo móvil.	Terminado.	2	008
2-008-5	Diseñar e implementar pantalla para el módulo de videoconferencia en el aplicativo móvil.	Terminado.	2	008
2-011-6	Diseño e implementación del servicio y lógica del de inicio de sesión para doctores mediante el directorio activo.	Terminado.	2	011
2-011-7	Diseñar e implementar vista para el módulo de inicio de sesión en el portal web.	Terminado.	2	011
2-007-8	Diseñar e implementar vista para el módulo de listado de emergencias en el portal web.	Terminado.	2	007
2-008-9	Crear conexión mediante webRTC utilizando PeerJS.	Terminado.	2	008
2-008-10	Configuración de servidores TURN y STUN.	Terminado.	2	008
2-008-11	Tranmisión de audio y video mediante webRTC utilizando PeerJS	Terminado.	2	008

Tabla 19 – Sprint backlog 3.

Código Actividad	Actividad	Estado	Iteracion (Sprint)	Historia de usuario (HU)
3-009-1	Diseño e implementación del servicio y lógica del manejo de chats para el beneficiario.	Terminado.	3	009
3-009-2	Diseñar e implementar pantalla para el módulo de chats que liste los chats abiertos de un Doctor (Portal web).	Terminado.	3	009
3-009-3	Diseñar e implementar pantalla para el módulo de chats que permita la transferencia de mensajes (Portal web).	Terminado.	3	009
3-009-4	Implementar almacenamiento del historial de la conversación (Portal web).	Terminado.	3	009
3-009-5	Diseñar e implementar pantalla para el módulo de chats para generar la comunicación con un Doctor (Aplicativo móvil).	Terminado.	3	009
3-009-6	Diseñar e implementar pantalla para el módulo de chats que permita la transferencia de mensajes (Aplicativo móvil).	Terminado.	3	009
3-009-7	Implementar almacenamiento del historial de la (Aplicativo móvil).	Terminado.	3	009
3-009-8	Implementar restricciones para manejo de múltiples conexiones en el Doctor (Portal web).	Terminado.	3	009

Tabla 20 – Sprint backlog 4.

Código Actividad	Actividad	Estado	Iteracion (Sprint)	Historia de usuario (HU)
4-009-9	Implementar carga de archivos adjuntos en el módulo de Chat (Portal web).	Terminado.	4	009
4-009-10	Implementar validaciones sobre formato y peso del archivo en el módulo (Portal web).	Terminado.	4	009
4-009-11	Implementar procesamiento de archivo para su envío (Portal web).	Terminado.	4	009
4-009-12	Implementar recepción de archivo y almacenamiento en dispositivo (Portal web).	Terminado.	4	009
4-009-13	Implementar carga de archivos adjuntos en el módulo de Chat (Aplicativo móvil).	Terminado.	4	009
4-009-14	Implementar validaciones sobre formato y peso del archivo en el módulo (Aplicativo móvil).	Terminado.	4	009
4-009-15	Implementar procesamiento de archivo para su envío (Aplicativo móvil).	Terminado.	4	009
4-009-16	Implementar recepción de archivo y almacenamiento en dispositivo (Aplicativo móvil).	Terminado.	4	009

Apéndice B – Artefactos del desarrollo

Sprint 1

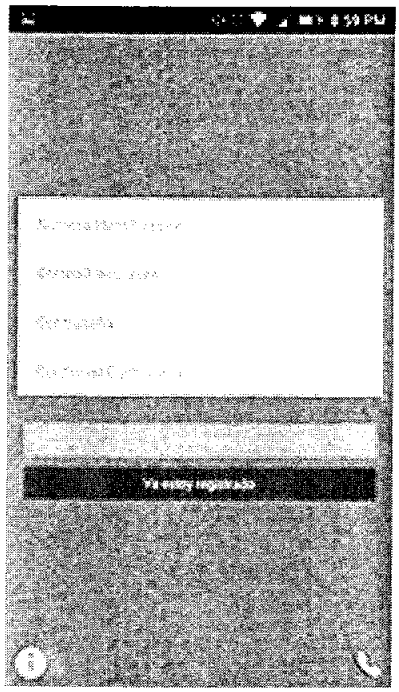

Figura 7 – Pantalla de registro (Aplicativo móvil)

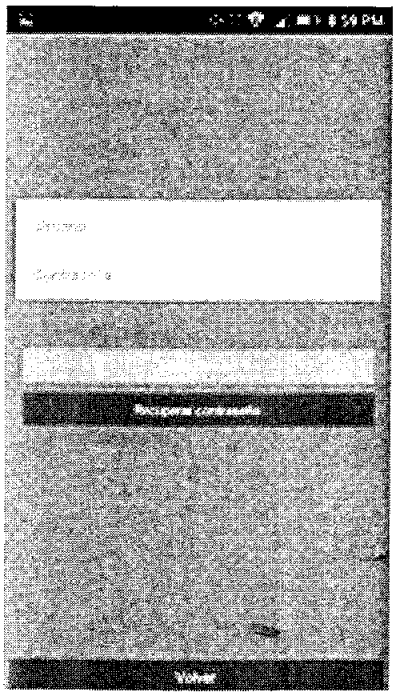

Figura 8 – Pantalla de inicio de sesión (Aplicativo móvil)

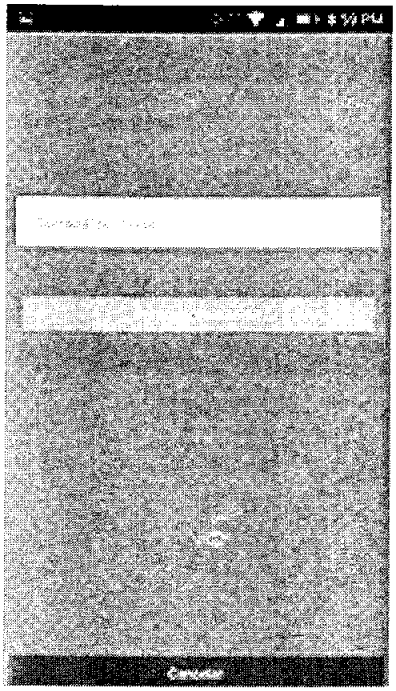

Figura 9 – Pantalla de recuperación de contraseña (Aplicativo móvil)

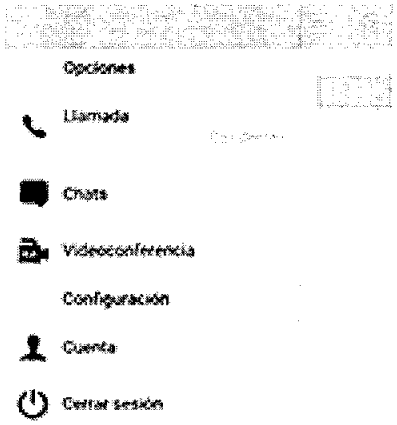

Figura 10 – Menú de navegación (Aplicativo móvil)

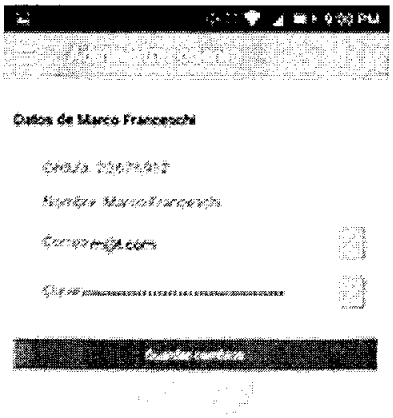

Figura 11 – Pantalla para administrar datos de cuenta de usuario (Aplicativo móvil)

Sprint 2

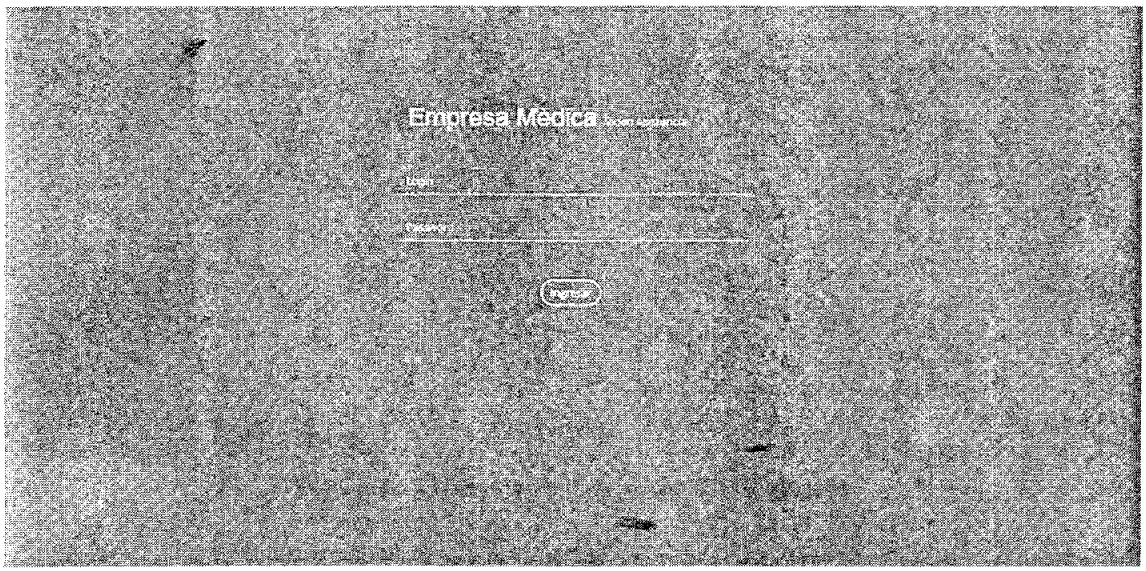

Figura 12 – Vista de inicio de sesión (Portal web)

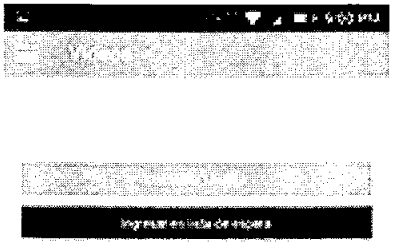

Figura 13 – Pantalla de videoconferencia (Aplicativo móvil)

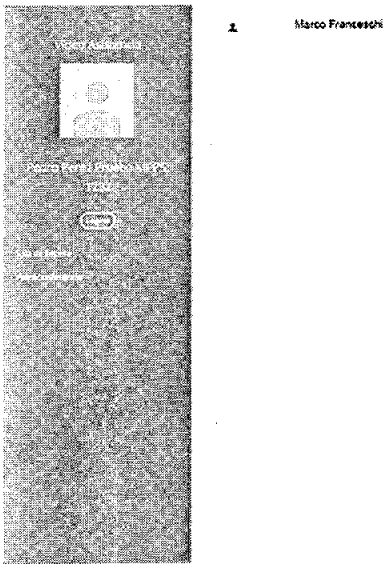

Figura 14 – Vista de lista de espera (Portal web)

Figura 15 – Vista de video conferencia en progreso (Portal web)

Figura 16 – Pantalla de video conferencia en progreso (Aplicativo móvil)

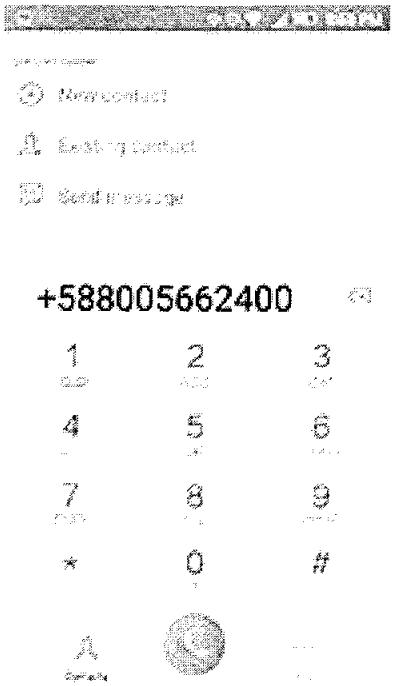

Figura 17 – Pantalla para llamada a *call center* (Aplicativo móvil)

Sprint 3

Figura 18 – Pantalla de nuevo chat (Aplicativo móvil)

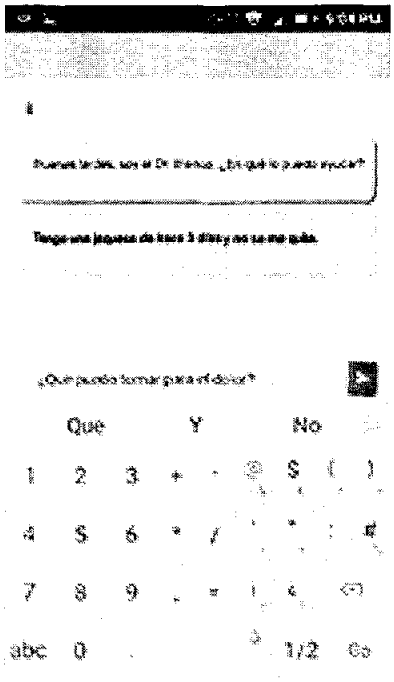

Figura 19 – Pantalla de chat (Aplicativo móvil)

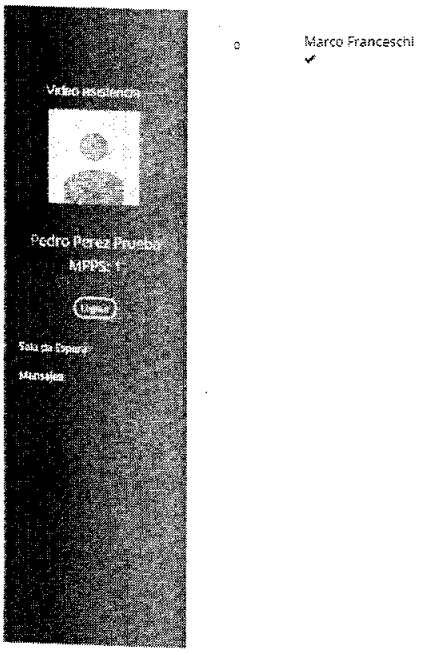

Figura 20 – Vista de listar *chats* (Portal web)

Figura 21 – Vista de chat (Portal web)

Sprint 4

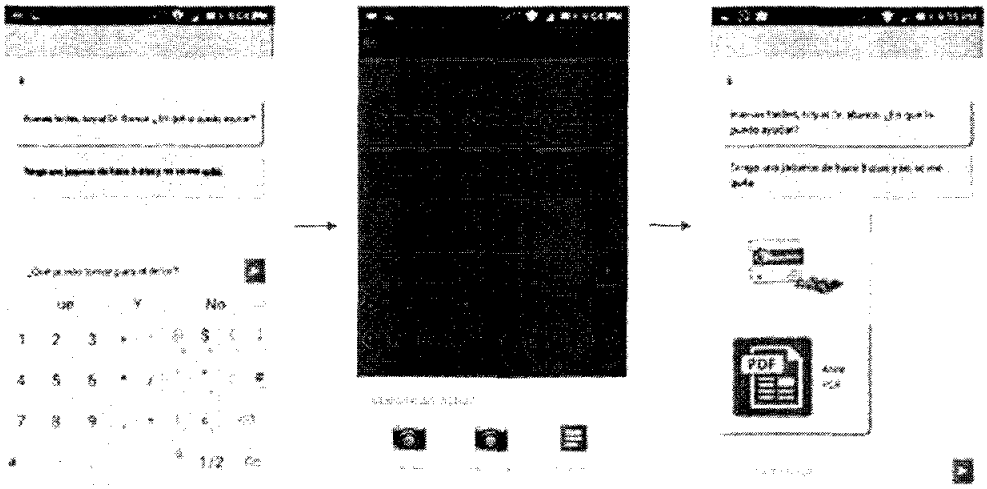

Figura 22 – Proceso de transferencia de archivos desde Aplicativo móvil

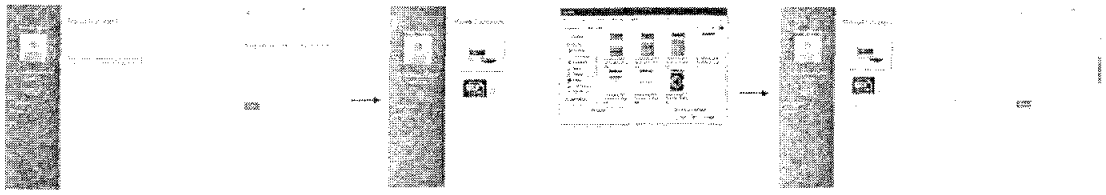

Figura 23 – Proceso de transferencia de archivos desde Portal web