



Desarrollo de aplicación Android: E-Health

Alexander Cabezas Medina

Máster en desarrollo de aplicaciones para dispositivos móviles

Pau Dominkovics Coll

Carles Garrigues

06/06/2018

Dedicado a mi esposa, sin ti ésta travesía jamás podría haber sido posible.

© Alexander Cabezas Medina

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	Desarrollo de Aplicación Android: E-Health
Nombre del autor:	Alexander Cabezas Medina
Nombre del consultor:	Pau Dominkovics Coll
Fecha de entrega (mm/aaaa):	06/2018
Titulación:	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>

Resumen del Trabajo (máximo 250 palabras):

Actualmente, los avances tecnológicos han permitido la democratización de muchos servicios a los cuales algunas poblaciones tendrían un acceso limitado por problemas de logística, como pueden ser de distribución de insumos o materiales, de personal etc. Entre esos servicios se encuentra la medicina.

Este proyecto tiene como objetivo principal proponer una solución al problema de hacer llegar a toda la población un servicio de medicina de calidad sin importar su localización geográfica o limitaciones de movimiento. Para lograr este objetivo se plantea el desarrollo de un sistema conformado por dos aplicaciones para dispositivos son sistema operativo android, una para los pacientes y otra para los doctores que van a contestar las consultas, además de un backend.

La metodología utilizada para el desarrollo de este proyecto ha sido SCRUM. También se ha utilizado diseño centrado en el usuario para determinar las necesidades y características de los usuarios que se quieren cubrir.

El desarrollo de ambas aplicaciones fue realizado de forma nativa con AndroidStudio como ambiente desarrollo y el SDK de Android, el backend fue desarrollado con nodejs y mongoDB como manejador de base de datos.

Por último, el resultado es un producto funcional y probado el cual cumple con los objetivos planteados al comienzo del proyecto.

Abstract (in English, 250 words or less):

Nowadays the technological improvements have allowed mankind to gain

access to many services to which many populations wouldn't have access or limited access due to logistic problems, as material or personal distribution. Among the services that can be difficult to distribute are the medical services.

The main goal of this project is to provide a solution to the high quality medical services access problem due to geographical or physical problems. To achieve the proposed goal, this project has come up with the development of a system that will have three parts, two mobile applications for mobile devices with android OS and a backend application that will be used by the mobile applications to save and retrieve information.

Both mobile applications are native applications and were developed using AndroidStudio which is the standard tool to develop android OS oriented applications. The backend application was developed using javascript programming language and the nodejs framework, also mongoDB was use as the DBMS.

The created product is a completely functional system that fulfill the main goal of this project.

Palabras clave (entre 4 y 8):

Android, Telemedicina, Dispositivos móviles

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo	2
1.3 Enfoque y método seguido.....	4
1.4 Planificación del Trabajo	7
1.5 Breve resumen de productos obtenidos.....	10
1.6 Breve descripción de los otros capítulos de la memoria	11
2. Análisis, diseño y arquitectura.....	12
2.1 Diseño centrado en el usuario.....	12
2.1.1 Usuarios y contextos de uso	12
2.1.1.1 Perfiles de usuario	13
2.1.2 Diseño conceptual.....	15
2.1.2.1 Definición de personas.....	15
2.1.2.2 Escenarios de uso.....	17
2.1.2.3 Flujos de interacción	22
2.1.3 Prototipado.....	24
2.1.4 Evaluación.....	30
2.2 Diseño técnico.....	31
2.2.1 Casos de uso	31
2.2.2 Modelo de datos.....	42
2.2.3 Diseño de la arquitectura	43
2.2.3.1 Arquitectura de la aplicación	43
2.2.3.2 Vista física.....	44
3. Implementación.....	45
3.1 Herramientas y APIS utilizadas.....	45
3.1.1 Android.....	45
3.1.2 AndroidStudio.....	45
3.1.3 Gradle	45
3.1.4 NodeJS	46
3.1.5 MongoDB	46
3.2 Aspectos relevantes del desarrollo	46
3.2.1 Proyecto Android.....	46
3.2.1.1 Implementación de interfaces de usuario.....	48
3.2.3 Servidor.....	48
3.3 Pruebas.....	53
4. Conclusiones.....	55
5. Glosario.....	57
6. Bibliografía	58
7. Anexos	59

Lista de figuras

Ilustración 1. Mend Telemedicine	4
Ilustración 2. Evisit Telehealth	5
Ilustración 3. Amwell	5
Ilustración 4. Diagrama de flujo para pacientes	23
Ilustración 5. Diagrama de flujo para doctores	24
Ilustración 6. Paquetes de casos de uso	32
Ilustración 7. Gestión de usuarios.	32
Ilustración 8. Gestión de citas	33
Ilustración 9. Video consulta	33
Ilustración 10. Modelo de datos	42
Ilustración 11. Vista física de las aplicaciones	44
Ilustración 12. Distribución de versiones de android	47
Ilustración 13. Estructura de paquete login	47
Ilustración 14. Estructura de paquete para el registro	47
Ilustración 15. Estructura del sdk del paciente	48
Ilustración 16. Modelo de lista de espera	49
Ilustración 17. Modelo de especialidades	49
Ilustración 18. Modelo de citas	50
Ilustración 19. Modelo de doctor	50
Ilustración 20. Modelo de paciente	51
Ilustración 21. Endpoints de doctores.	52
Ilustración 22. Endpoints de pacientes	53

1. Introducción

1.1 Contexto y justificación del Trabajo

Aunque en la actualidad la tecnología avanza a un ritmo vertiginoso, y con esos avances se ha mejorado la calidad de vida de los seres humanos en una innumerable cantidad de formas, algunos servicios básicos todavía se encuentran fuera del alcance de algunas personas bien sea por limitaciones económicas, geográficas o tecnológicas. Como ejemplo tenemos el acceso a la medicina.

Es cierto que los avances en medicina en los últimos años han aumentado la esperanza de vida de las personas, curan enfermedades que antes se consideraban incurables y para las que aún no se tiene una cura cada día se ofrecen tratamientos más efectivos, sin embargo, estos avances no están al alcance de todos, o al menos, no de una forma fácilmente accesible.

Podemos mencionar algunos casos que demuestran lo anterior, como pacientes de movilidad reducida o que les cuesta desplazarse como pueden ser ancianos o personas en recuperación, también podríamos incluir personas que deben desplazarse por largas distancias para obtener atención médica, o esperar mucho tiempo para poder ser atendidos o evaluados por un especialista.

Lo anteriormente mencionado, produce un detrimento en la salud y calidad de vida de todas esas personas, ya que son un sector de la población que se encuentran dentro de los grupos de riesgo dada su condición.

Para mitigar los efectos de este problema, mejorar la calidad de vida de las personas dentro de esos grupos de riesgo y facilitar el acceso a la salud sin importar la ubicación geográfica, se propone una solución basada en dispositivos móviles que permita a estos pacientes tener vídeo consultas con los especialistas de cualquier área de la medicina sin necesidad de moverse de su lugar, además de esto y para hacer más accesible el uso de la aplicación se aprovechara la tecnología de los dispositivos para que los pacientes puedan utilizar la aplicación mediante comandos de voz.

También, se plantea en este trabajo realizar el desarrollo de todo el sistema como un caso real, como si se tuvieran clientes, un equipo multidisciplinario, fechas de entrega etc.

Tomando en cuenta que este trabajo será realizado como un caso real, se plantea el uso de SCRUM como metodología de desarrollo. Esto se debe a que las metodologías ágiles se adaptan bastante bien a los ciclos de desarrollo de los productos de software y en un ambiente real donde se maneja una alta

variabilidad en cuanto a las estimaciones de tiempo para las funcionalidades a desarrollar además de tener flexibilidad para manejar los cambios en requisitos tecnológicos o capital humano del proyecto.

Como este trabajo final de master cuenta con fechas delimitadas, fases definidas por cada entrega parcial las cuales tienen unos objetivos por cumplir, se adaptará la metodología para cumplir con estos hitos.

1.2 Objetivos del Trabajo

En este apartado se describen los objetivos de este trabajo final de master, los requerimientos funcionales y no funcionales de las aplicaciones móviles y demás partes del sistema necesarios para este proyecto.

El objetivo principal de este trabajo final de master es crear un producto funcional que permita a pacientes tener vídeo consultas con especialistas de cualquier área de la medicina y que la aplicación destinada a los pacientes pueda ser utilizada sin necesidad de tocar la pantalla del dispositivo, para que de esta manera pueda ser utilizada por personas mayores, con movilidad o visión reducida.

Objetivos:

- Desarrollar una aplicación para dispositivos móviles con sistema operativo android que permita a los pacientes tener vídeo consultas con especialistas de cualquier área de la salud.

- Implementar un módulo para la gestión de usuarios.
 - o Registro de paciente, en el cual se obtendrá la información personal y parte de su historial clínico.
 - o Recuperar contraseña.
 - o Manejo de sesión.
 - o Actualizar datos personales y de historia médica.
- Implementar un módulo para la vídeo conferencia.
 - o Buscar especialidad entre las disponibles.
 - o Realizar una vídeo consulta, en cual se comunique con el especialista seleccionado.
- Implementar un módulo para programar una cita.
 - o Poder programar una cita de una especialidad médica, en la cual se podrá seleccionar la fecha y hora para ser atendido.
 - o Reprogramar una cita a otro día u hora.
 - o Eliminar una cita programada.
- Implementar un módulo para calificar la atención recibida.
 - o Responder una encuesta sobre la atención recibida.

- Desarrollar una aplicación para dispositivos móviles con sistema operativo android que permita a los médicos contestar las vídeo llamadas realizadas por los pacientes.

- Implementar un módulo para la gestión de médicos.
 - o Registro de un médico, en el cual además de su información personal podrá indicar la(s) especialidad(es) que posee.
 - o Recuperar contraseña.
 - o Manejo de sesión.
 - Implementar un módulo para la vídeo conferencia.
 - o Realizar una vídeo consulta con un paciente.
 - o Ver pacientes en cola de espera a ser atendidos.
 - Implementar un módulo para responder a la solicitud de una cita de un paciente.
 - o Ver citas programadas por los pacientes.
- Desarrollar un backend que permita el manejo de la comunicación entre las aplicaciones, almacene los usuarios y su información médica.
- Implementar un módulo para la gestión de usuarios.
 - o Servicio web de registro de pacientes.
 - o Servicio web de registro de médicos.
 - o Recuperar contraseña.
 - o Manejo de sesión.
 - o Actualizar datos personales y de historia médica de un paciente.
 - o Actualizar datos personales y profesionales de un médico.
 - Implementar un módulo para la vídeo conferencia.
 - o Servicios web que permitan la realización de una video consulta.
 - o Servicio web que permita obtener los pacientes en la cola de espera para ser atendidos.
 - Implementar un módulo para programar una cita.
 - o Servicio web que permita programar una cita de una especialidad médica.
 - o Servicio web que permita obtener la lista de las citas programadas.
 - o Servicio web que permita programar una cita.
 - o Servicio web que permita eliminar una cita programada.
 - o Servicio web que permita reprogramar una cita.
 - Implementar un módulo para calificar la atención recibida.
 - o Servicio web que permita ver calificaciones obtenidas.

Requisitos no funcionales:

- La aplicación debe ser desarrollada para dispositivos con sistema operativo Android.

- El sistema deberá estar disponible en todo momento.
- El sistema deberá ser robusto, seguro y fiable.

1.3 Enfoque y método seguido

Actualmente ya existen algunos productos que ofrecen servicios de telemedicina basados en dispositivos móviles como

Mend Telemedicine. Aplicación que ofrece una plataforma para conectar pacientes con proveedores de salud, mediante mensajes, video conferencias. También puede recolectar información médica de los pacientes a través de terceros, se puede encontrar en <https://play.google.com/store/apps/details?id=com.mend.android.telemedicine&hl=es>. También se muestran algunas imágenes de su interfaz de usuario.

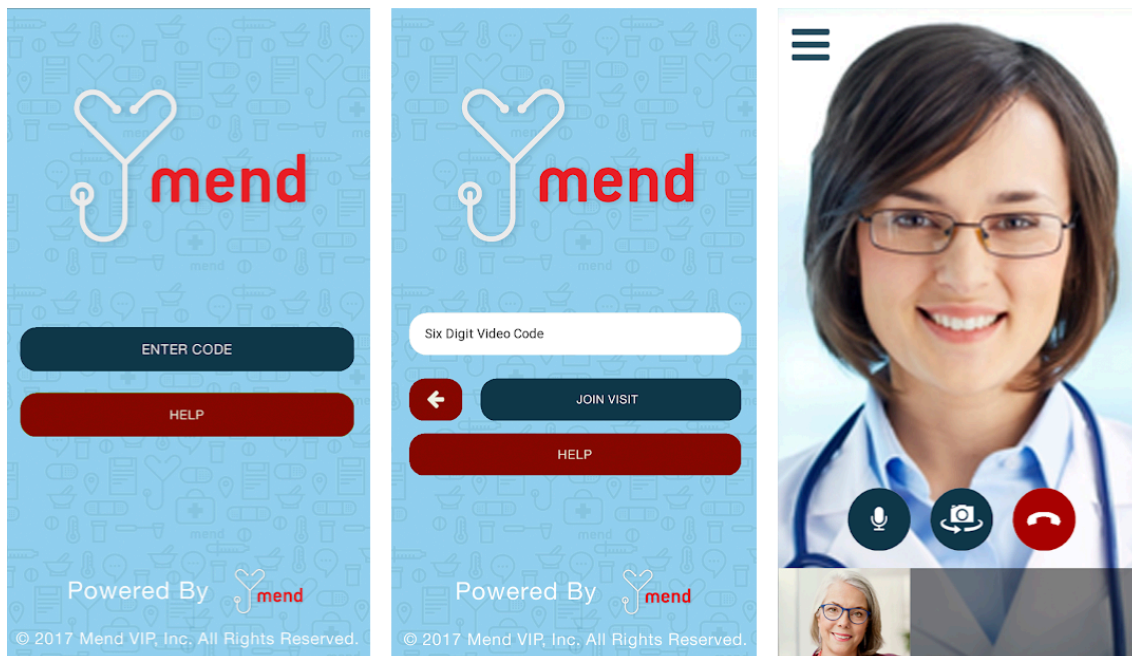


Ilustración 1. Mend Telemedicine

eVisit Telehealth. Aplicación que permite a un paciente poder ser atendido por un doctor de la lista de médicos disponibles en ese momento, dispone de funcionalidades de mensajería y video conferencia, además de poder hacer prescripciones de medicinas mediante otra plataforma llamada ePrescribe. Se puede encontrar en <https://play.google.com/store/apps/details?id=com.evisit.app>. También se colocan algunas imágenes de su interfaz gráfica.

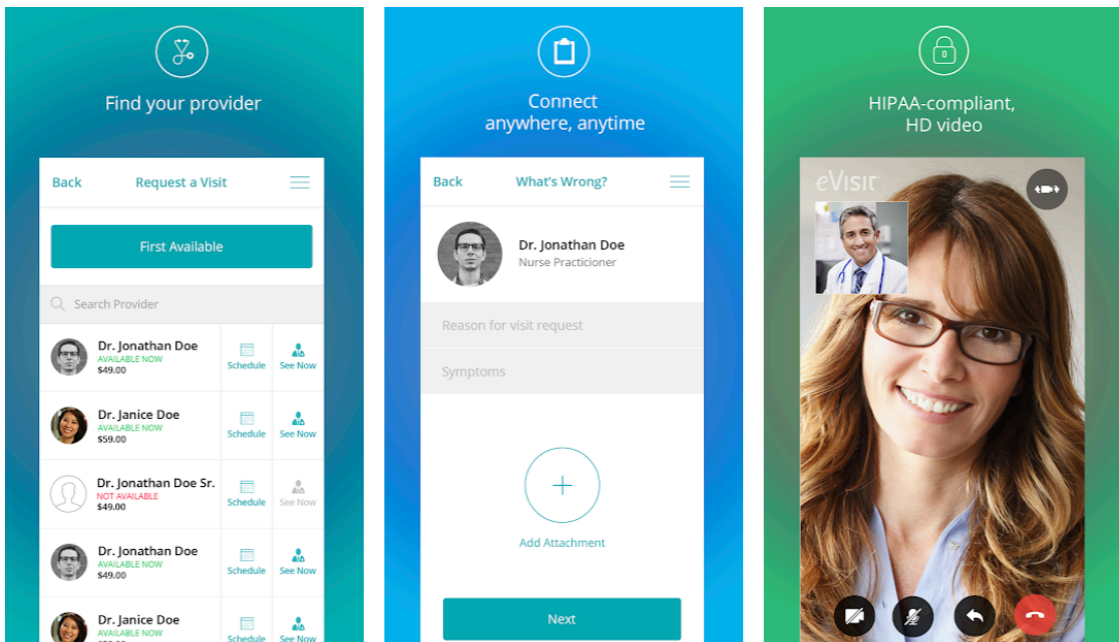


Ilustración 2. Evisit Telehealth

AmWell. Aplicación que permite a los pacientes tener video consultas con médicos que se encuentren disponibles o mediante citas programadas previamente y que funciona las 24 horas del día. Ofrece un amplio rango de servicios médicos como terapia, psiquiatría, nutrición, etc. Se puede encontrar en

<https://play.google.com/store/apps/details?id=com.americanwell.android.member.amwell&hl=en>. También se agregan algunas imágenes de la aplicación.

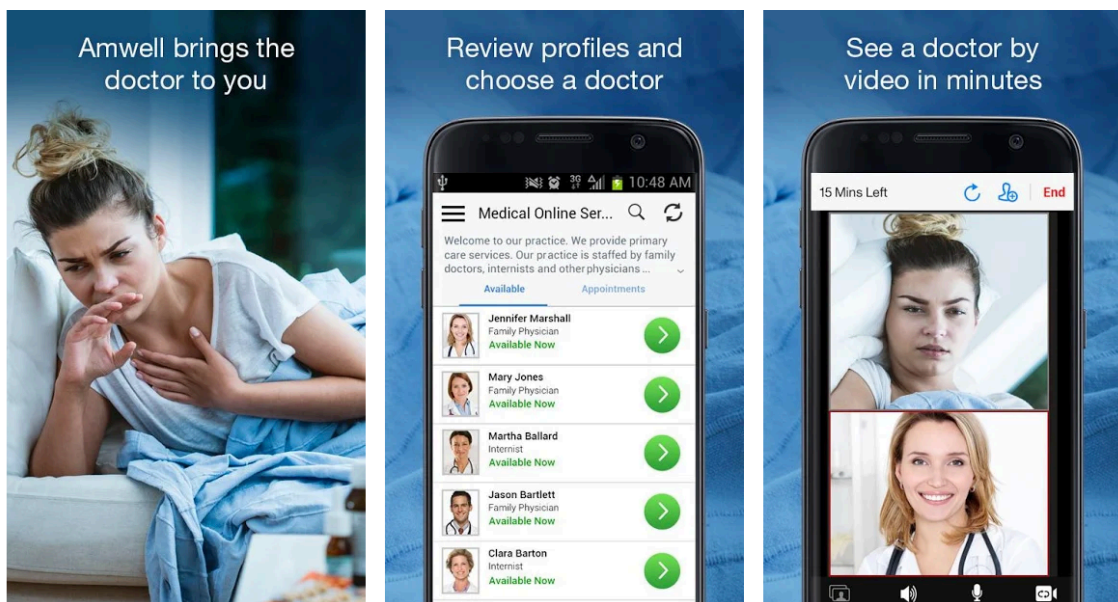


Ilustración 3. Amwell

Sin embargo, los productos descritos previamente no se enfocan en dar accesibilidad a las aplicaciones móviles que poseen para que puedan ser utilizadas por personas con visión y/o movilidad reducida. Por lo tanto, se plantea desarrollar un nuevo producto desde cero.

Al desarrollar un nuevo producto es posible adaptarlo a las necesidades y requerimientos de los pacientes que se quieren atender, por lo que esta estrategia es más efectiva para cumplir con los objetivos planteados.

Como se comentó en el apartado 1.1, la metodología elegida para el desarrollo de este proyecto, es SCRUM. Dicha metodología fue escogida ya que permite elaborar un producto de forma incremental con funcionalidades que se pueden ir utilizando una vez culminado cada sprint de desarrollo. Esto permite hacer pruebas y cambios en periodos muy cortos de tiempo.

Además de utilizar SCRUM como metodología de desarrollo también se utilizará el Diseño Centrado en el Usuario, lo cual permitirá enfocar todos los esfuerzos en cumplir con las necesidades de los usuarios de una forma progresiva e iterativa, donde en cada iteración se agregarán las funcionalidades de máximo valor para los usuarios.

Scrum como metodología de desarrollo cuenta con los siguientes eventos y artefactos.

- Eventos:

- Sprint: este es un evento con tiempo delimitado en el cual es donde se desarrolla el nuevo incremento al producto, para este proyecto los Sprint serán de 1 semana.
- Sprint Planning: este evento es donde se realiza la estimación de las tareas y se evalúa cuáles de ellos se van a desarrollar en un Sprint.
- Daily Scrum: este evento es una reunión diaria acotada a un máximo de 15 minutos en la cual cada miembro del equipo expone las tareas que realizara ese día y si tiene algún elemento que bloquee su trabajo.
- Sprint Review: este evento se revisa el nuevo incremento desarrollado durante el Sprint.
- Sprint Retrospective: este evento se utiliza para evaluar el desempeño del equipo durante el Sprint, se revisan los puntos de mejora del equipo para el siguiente Sprint.

- Artefactos:

- Product Backlog: este artefacto contiene todas las historias de usuario y tareas que deben ser desarrolladas para el producto y se encuentra organizado de tal manera que los elementos que agregan más valor al producto tengan prioridad para ser agregados a un Sprint.
- Sprint Backlog: este artefacto contiene todas las historias de usuario y tareas que se van a desarrollar durante el Sprint.
- Increment: este artefacto es el producto resultante del desarrollo de las historias de usuario en un sprint, lo que representa un incremento en las funcionalidades del producto, dándole a este mayor valor para los usuarios.

- Actores:

- Product Owner: este rol se encarga de mantener el backlog siempre con las funcionalidades que agreguen mayor valor al producto.
- Development team: es el equipo encargado de desarrollar cada incremento del producto
- Scrum master: es un facilitador para todo el equipo, ayuda a implementar la metodología, así como a la empresa a adaptarse a las metodologías ágiles.

Para este trabajo final de master se adaptará SCRUM para cumplir con los objetivos de cada entrega parcial. La duración de cada sprint será de una semana, ya que se considera ese es un tiempo apropiado para el desarrollo de cada una de las actividades y suficiente para generar un incremento del producto.

La capacidad de trabajo para cada sprint será de 18 horas, por lo que el sprint backlog se ajustará para tener una carga de trabajo que se pueda llevar a cabo en ese tiempo.

También, al contar con sprints de una semana, se ajustan las fechas de entrega de cada PEC al final de un sprint.

1.4 Planificación del Trabajo

Los recursos necesarios para el desarrollo de este Trabajo estarán divididos en tres tipos: hardware, software y capacidad humana, al ser un Trabajo Final de Master la capacidad humana es solo una persona que es el estudiante.

Los recursos de hardware para el desarrollo de este proyecto son:

- 1 ordenador portátil MacBook Pro de 13 pulgadas, con procesador intel i7 8 Gb de memoria Ram y 128 GB SSD.
- 1 teléfono inteligente BQ M5 con android 6.0.

Los recursos de software:

- AndroidStudio.
- Android SDK.
- Microsoft Office.

El plan de trabajo desarrollado cuenta con 5 fases, las cuales se basan en el modelo de evaluación continua propuesto por la OUC. Cada fase finaliza con un hito el cual se cumple al realizar la entrega de cada una de las PECs.

- Fase 0: Definición del proyecto.
 - o Definición del proyecto.
 - o Planteamiento de la idea a desarrollar.
 - o Documentación de la propuesta.

En la fase 0 se requiere pensar y redactar la propuesta del trabajo final de master para ser presentada ante el consultor y este la debe aprobar para poder ser usada en el proyecto.

- Fase 1: Plan de trabajo.
 - o Contexto y motivación.
 - o Objetivos y requisitos.
 - o Enfoque y metodología.
 - o Planificación del proyecto.
 - o Documentación de la PEC1.
 - o Entrega PEC1.

La fase 1 es la fase inicial del trabajo final del master en la cual se extiende la idea de la propuesta inicial al el contexto y justificación de la misma, los objetivos del proyecto, la metodología que se utilizara durante el proceso de desarrollo, se realizara la planificación basada en el tiempo disponible y los hitos requeridos por la evaluación continua y finalmente se entrega la primera versión de la memoria del trabajo.

- Fase 2: Análisis, diseño y arquitectura.
 - o Definición de usuarios.
 - o Diseño conceptual.
 - o Casos de uso.
 - o Creación de prototipos.
 - o Diseño de arquitectura.
 - o Documentación PEC2.
 - o Entrega PEC2.

En esta fase realiza todo el diseño relacionado con el proyecto, comenzando con el diseño centrado en el usuario al recopilar información sobre requerimientos, forma de uso etc. También se realiza el diseño técnico del sistema que se va a implementar, como modelos de datos, arquitectura del sistema y díselo del api de comunicación entre las aplicaciones y el backend. Además, se realizan los prototipos de baja y alta fidelidad de las aplicaciones.

- Fase 3: Implementación.
 - o Preparación del entorno.
 - o Desarrollo de backend.
 - o Desarrollo de aplicación paciente.
 - o Desarrollo de aplicación doctor.
 - o Documentación PEC3.
 - o Entrega PEC3.

En esta fase se realiza desarrollo de todo el sistema de forma iterativa e incremental, agregando una nueva parte del sistema en cada sprint. Al inicio se prepara el ambiente de desarrollo y luego se comienza con el backend, seguido se realiza el desarrollo de las aplicaciones para los pacientes y médicos.

- Fase 4: Entrega final
 - o Elaboración de la presentación.
 - o Grabación vídeo presentación.
 - o Documentación entrega final.
 - o Entrega final.

En esta última fase se prepara la entrega final del trabajo, en la cual se realiza una presentación con los detalles más importantes del trabajo como decisiones tomadas durante el proceso de desarrollo, conclusiones y resultados obtenidos. Luego se grabará un video utilizando dicha presentación como base además de mostrar el funcionamiento del sistema. Finalmente se llevará a cabo la entrega final del trabajo.

En las tablas 1, 2, 3 y 4 se muestra la planificación temporal de cada uno de los sprints necesarios para cumplir con los objetivos de cada uno de los hitos mencionados anteriormente. En cada tabla se muestran las tareas por cada sprint y su estimación. Como se mencionó en el apartado anterior, cada sprint contará con una duración de una semana, y se estima que se tendrá una capacidad de 18 horas disponibles por cada sprint. También la fecha de entrega de cada una de las PECs, se realiza al finalizar un sprint.

PEC 1

Tabla 1

Sprint	Fecha de inicio	Fecha fin	Tareas	horas
1	21/02/2018	27/02/2018	Definición del proyecto	6
			Planteamiento de la idea a desarrollar	6
			Documentación de la propuesta	6
2	28/02/2018	06/03/2018	Contexto y motivación	6
			Objetivos y requisitos	6
			Enfoque y metodología	6
3	07/03/2018	14/03/2018	Planificación del proyecto	6
			Documentación de la PEC1	12
			Entrega PEC1	

PEC 2

Tabla 2

Sprint	Fecha de inicio	Fecha fin	Tareas	horas
4	15/03/2018	21/03/2018	Definición de usuarios	9

			Casos de uso	9
5	22/03/2018	28/03/2018	Diseño conceptual	9
			Creación de prototipos	9
6	29/03/2018	04/04/2018	Diseño de arquitectura	9
			Documentación PEC2	9
			Entrega PEC2	

PEC 3

Tabla 3

Sprint	Fecha de inicio	Fecha fin	Tareas	horas
7	05/04/2018	11/04/2018	Preparación del entorno	4
			Desarrollo de backend	14
8	12/04/2018	18/04/2018	Desarrollo de aplicación paciente	18
9	19/04/2018	25/04/2018	Desarrollo de aplicación paciente	18
10	26/04/2018	02/05/2018	Desarrollo de aplicación doctor	18
11	03/05/2018	09/05/2018	Desarrollo de aplicación doctor	18
12	10/05/2018	16/05/2018	Documentación PEC3	18
			Entrega PEC3	

Entrega final

Tabla 4

Sprint	Fecha de inicio	Fecha fin	Tareas	horas
13	17/05/2018	23/05/2018	Elaboración de la presentación	18
14	24/05/2018	30/05/2018	Grabación vídeo presentación	18
15	31/05/2018	06/06/2018	Documentación entrega final	18
			Entrega final	

1.5 Breve resumen de productos obtenidos

Los productos entregables generados por este Trabajo son:

- El sistema desarrollado, el cual se encuentra conformado por dos aplicaciones móviles, una para los pacientes y otra para el personal de salud y la aplicación del backend. Se entregarán tanto los apk como código Fuente de ambas aplicaciones y del backend.
- Una memoria del Trabajo final de master, donde se encuentra explicado todo el proyecto realizado.
- Presentación del proyecto, en la cual se hace una síntesis sobre las partes más importantes del proyecto.

- Un vídeo de presentación, en el cual se explica el funcionamiento de las distintas partes del proyecto y se realiza una demo de las aplicaciones.
- Anexos, como manual de usuario de las aplicaciones, manual de instalación de las aplicaciones, etc.

1.6 Breve descripción de los otros capítulos de la memoria

Los capítulos posteriores de este Trabajo son:

- El capítulo de diseño y arquitectura, en el cual se expone todo lo referente a la creación de las interfaces de usuario, usabilidad y diseño de interacción, además del diseño técnico, de la arquitectura de las aplicaciones y definición de modelos de datos y del API para la comunicación entre las aplicaciones móviles y la aplicación del backend.
- El capítulo de implementación, en el cual se expone el proceso de desarrollo del producto, incluyendo las tecnologías y herramientas utilizadas, detalles relevantes del desarrollo y las pruebas para validar el correcto funcionamiento de las aplicaciones.
- Conclusiones, en las cuales se expone las lecciones aprendidas, un análisis sobre los procesos utilizados durante la implementación y la metodología elegida.

2. Análisis, diseño y arquitectura

2.1 Diseño centrado en el usuario

El diseño centrado en el usuario es un enfoque en el cual el proceso de diseño está dirigido por la información recolectada de los usuarios cuyas necesidades se plantean resolver con el nuevo producto.

El diseño centrado en el usuario es un proceso cíclico, en el que cada etapa coloca en el centro de atención de los objetivos que pretende satisfacer el producto, siempre guiados por las necesidades de los usuarios. En el final de cada ciclo se obtiene nueva información de los usuarios lo que permite mejorar y adaptar los elementos para crear un nuevo incremento.

Según la norma ISO 13407 se tienen cuatro fases para el diseño centrado en el usuario.

- Entender y especificar el contexto de uso: en esta fase se identifican a las personas a las que se dirige el producto, para que lo van a usar y en cuales condiciones.
- Especificar requisitos: en esta fase se identifican los objetivos de los usuarios que se deben satisfacer con el producto.
- Producir soluciones de diseño: en esta fase es donde se desarrollan de forma iterativa prototipos de diseño desde baja fidelidad hasta alta fidelidad con el diseño final.
- Evaluación: esta es la fase más importante ya que es donde se valida que el diseño satisface los objetivos, o si por el contrario existen problemas con el mismo o con la usabilidad.

2.1.1 Usuarios y contextos de uso

En esta primera etapa del diseño centrado en el usuario se busca obtener las características, necesidades y objetivos de los usuarios. Para lograr esto se utilizan una variedad de métodos de indagación en los cuales se busca obtener datos directamente de los usuarios, como pueden ser encuestas, entrevistas, etc.

En este trabajo no se utilizarán los métodos más comunes para recabar información directamente de los usuarios ya que es complicado obtener datos directamente del público objetivo y no se plantea realizarlos a otro público ya que se obtendrían datos poco fiables. Por esto se decidió utilizar el método de benchmarking.

En el método de benchmarking se analizan productos similares o que son la competencia del producto que se quiere desarrollar, esto permite conocer a los usuarios y sus expectativas, saber las tendencias del mercado para este tipo de producto, saber cuáles son las funcionalidades más usadas y cuáles no, estudiar las interfaces graficas que estos otros productos ya ofrecen, etc.

En la introducción de este trabajo ya se analizaron algunas aplicaciones similares al producto que se quiere diseñar en este trabajo, por lo que se utilizará la información obtenida de dicho análisis para obtener los objetivos a satisfacer por el producto.

De las aplicaciones analizadas en el primer capítulo de este trabajo se obtuvo información muy relevante sobre las funcionalidades más comunes ofrecidas por este tipo de aplicaciones entre las cuales tenemos:

- La vídeo consulta, es una funcionalidad fundamental para que el médico y el paciente no solo se comuniquen en tiempo real, si no para que también el médico pueda evaluar de forma visual al paciente.
- Poder ser atendido por un médico que se encuentre disponible al momento de realizar la vídeo llamada y que el tiempo de espera sea lo menor posible, y siempre mostrar al paciente en una sala de espera virtual en la cual se le indique el tiempo estimado de espera.
- Ofrecer múltiples especialidades médicas dentro de la aplicación, esto permite atender a la mayor cantidad de pacientes posibles ya que se pueden cubrir una gran cantidad de enfermedades.
- Programar una cita con un médico, esta funcionalidad permite ser atendido por un médico especialista en un día y hora determinado lo cual minimiza el tiempo de espera y da gran flexibilidad a los pacientes para manejar su tiempo, además permite programar citas de seguimiento en el caso de que el tratamiento lo requiera.

2.1.1.1 Perfiles de usuario

Los perfiles de usuarios, son agrupaciones creadas en base a sus características, como pueden ser objetivos, necesidades, expectativas, etc. Los distintos perfiles de usuarios identificados en esta fase son:

Perfil de usuario paciente	
Características del perfil	<p>Este perfil estaría formado por todos aquellos usuarios que utilizan la aplicación para obtener un Servicio de consultas médicas por medio de la aplicación, ya que por motivos de tiempo, distancia o dificultad de movimiento les es muy difícil dirigirse de forma física hasta donde se encuentra el médico.</p> <p>El objetivo fundamental para estos usuarios es poder tener acceso a un Servicio de salud de calidad reduciendo o eliminando por completo el desplazamiento y el tiempo de espera para ser atendido por un profesional de la salud.</p>
Contexto de uso	Los usuarios que pertenecen a este

	grupo usaran la aplicación para buscar la especialidad médica que necesitan y poder tener una consulta en ese momento o programar una cita para un momento conveniente.
Tareas	<ul style="list-style-type: none"> - Registrarse en la aplicación - Iniciar sesión - Buscar una especialidad médica. - Tener una consulta. - Programar una cita para una consulta
Listado de características	<ul style="list-style-type: none"> - Poder ver las Especialidades médicas ofrecidas. - Poder pedir una cita en el momento en el que se inicia sesión en la aplicación. - Poder programar una cita con un especialista para otro momento

Perfil de usuario medico	
Características del perfil	<p>Este perfil estaría formado por todos aquellos usuarios que son médicos y se registran en la aplicación como tales para ofrecer sus Servicios.</p> <p>El objetivo fundamental de este perfil es poder ofrecer sus servicios por medio de la aplicación para ser capaces de atender pacientes con movilidad reducida. O que se encuentren a una gran distancia.</p>
Contexto de uso	Los usuarios que pertenezcan a este perfil utilizaran la aplicación para atender citas en el momento en el que el paciente lo necesite o citas programadas por pacientes.
Tareas	<ul style="list-style-type: none"> - Registrarse en la aplicación. - Iniciar sesión. - Atender una cita de un paciente en espera. - Atender una cita programada por un paciente
Listado de características	<ul style="list-style-type: none"> - Poder atender una cita de un paciente que se encuentra en espera. - Poder atender una cita programada por un paciente.

2.1.2 Diseño conceptual

En esta fase se utiliza la información obtenida de la etapa anterior para resumir y transmitir los datos y así, de esta forma, tomar decisiones en cuanto al diseño final del producto que debe satisfacer los objetivos planteados.

2.1.2.1 Definición de personas

Esta técnica se basa en la creación de personajes que detallan a posibles usuarios del producto a partir de la información obtenida en las fases anteriores, de esta manera siempre mantenemos las necesidades de los usuarios en el centro del proceso de diseño.

Personaje 1	
Pedro Gutiérrez	Pedro tiene 32 años, es casado con un hijo de 2 años, es ingeniero informático y Trabaja desarrollando software en un banco, gana 40 mil euros al año y vive en Madrid. Entre el Trabajo y su familia no tiene mucho tiempo libre. Sufre de mucho estrés y esto le hace asistir periódicamente a terapia para poder manejarlo.
Conocimientos tecnológicos	Al ser un ingeniero informático se desenvuelve perfectamente con ordenadores y dispositivos móviles ya que gran parte de su día se encuentra trabajando con ellos, utiliza su dispositivo móvil con regularidad tanto para temas laborales, como personales y ocio.
Objetivos	<ul style="list-style-type: none">- Poder asistir a terapia, pero sin invertir tanto tiempo en desplazamientos ya que debido a estos pierde mucho tiempo que puede invertir con su familia.- Poder asistir a terapia desde su propia casa u oficina
Necesidades	<ul style="list-style-type: none">- Poder asistir a terapia sin invertir tanto tiempo en desplazamientos,- Poder programar sus citas de terapia con antelación.

Personaje 2	
Carla Pérez	Carla tiene 35 años, es soltera y sin

	hijos. Trabaja en una oficina bancaria cerca de su casa como cajera. Tiene movilidad reducida debido a un accidente de automóvil que sufrió durante su infancia. Este accidente donde tuvo fracturas en varios huesos de su pierna derecha le impide caminar con normalidad, por lo que desplazarse le cuesta Trabajo.
Conocimientos tecnológicos	Está familiarizada con el uso de teléfonos inteligentes y tabletas ya que usa bastante las redes sociales, además al usar sistemas informáticos en su trabajo diario, está bastante familiarizada con los ordenadores.
Objetivos	<ul style="list-style-type: none"> - Poder asistir a consultas médicas con el menor desplazamiento posible ya que le cuesta movilizarse debido a sus lesiones en la pierna derecha.
Necesidades	<ul style="list-style-type: none"> - Desplazarse lo menos posible para ir a sus citas médicas

Personaje 3	
Pedro Pérez	Pedro tiene 30 años, vive con sus padres ya que sufre de una enfermedad muscular degenerativa lo cual lo obliga a estar en cama, o a desplazarse en una silla de ruedas con ayuda. Debido a su enfermedad no le es posible desplazarse por sí mismo y tampoco coger objetos con las manos. Necesita de citas médicas de seguimiento y terapia de forma recurrente.
Conocimientos tecnológicos	Esta bastante familiarizado con los dispositivos móviles como teléfonos inteligentes y tabletas, ya que debido a su enfermedad ha pasado mucho tiempo en casa donde su principal forma de entretenimiento han sido estos dispositivos y ordenadores.
Objetivos	<ul style="list-style-type: none"> - Poder tener las citas médicas de seguimiento sin necesidad de desplazarse. - Poder tener citas médicas por sí mismo si es necesario.
Necesidades	<ul style="list-style-type: none"> - Tener citas médicas des su

	<p>hogar para evitar desplazamientos innecesarios.</p> <ul style="list-style-type: none"> - Ser capaz de tener una cita médica por sí mismo.
--	---

Personaje 4	
Carlos Gonzales	Carlos tiene 40 años es médico internista. Está casado y tiene un hijo. En las mañanas trabaja como médico de la Seguridad social y en las tardes ofrece consultas privadas.
Conocimientos tecnológicos	Es bastante hábil con la tecnología, está familiarizado con ordenadores y dispositivos móviles ya que se encuentra muy interesado en las nuevas tecnologías.
Objetivos	<ul style="list-style-type: none"> - Poder ofrecer sus consultas privadas desde casa. - Poder ayudar a pacientes que se encuentren a grandes distancias
Necesidades	<ul style="list-style-type: none"> - Ayudar a más pacientes, aunque se encuentren a una distancia considerable. - Ayudar que los pacientes que les cueste desplazarse puedan ser atendidos.

2.1.2.2 Escenarios de uso

En este apartado se desarrollarán los posibles escenarios de uso, los cuales van a describir una situación en la cual alguno de los personajes definidos en el apartado anterior hace uso del producto con unos objetivos determinados. Con estos escenarios se podrá observar la forma en que los personajes pueden interactuar con la aplicación para cumplir sus objetivos.

Escenario 1 Registrarse en la aplicación como paciente	
Perfil de usuario	Paciente
Personaje	Pedro Gutiérrez
contexto	Son las 14 horas y pedro quisiera tomar una cita médica ya que se siente un mal, sin embargo, no quiere tener que pedir permiso en el Trabajo para asistir, así que piensa en alguna otra opción donde no tenga que desplazarse y por eso busca una aplicación que el permita tener una cita médica mediante una vídeo conferencia,
objetivo	Registrarse en la aplicación

Funcionalidades	Registro de un paciente
Desarrollo de tareas	<ul style="list-style-type: none"> - Buscar una aplicación en la Play Store. - Descargar la aplicación. - Registrarse en la aplicación colocando sus datos personales y datos médicos básicos.

Escenario 2 Registrarse en la aplicación como medico	
Perfil de usuario	Médico
Personaje	Carlos Gonzales
contexto	Carlos quiere comenzar a atender sus pacientes a distancia, ya que algunos de ellos tienen movilidad reducida y otros viven muy lejos, por lo que decidí buscar alguna plataforma que le permita realizar esto, especialmente si esta plataforma es por medio de dispositivos móviles.
objetivo	Registrarse en la aplicación
Funcionalidades	Registro de usuario como medico
Desarrollo de tareas	<ul style="list-style-type: none"> - Buscar una aplicación en la Play Store. - Descargar la aplicación. - Registrarse en la aplicación colocando sus datos personales y datos profesionales.

Escenario 3 Ver lista de especialidades que ofrece la aplicación	
Perfil de usuario	Paciente
Personaje	Pedro Gutiérrez
contexto	Pedro quiere tomar una cita con un psiquiatra para seguir con su terapia de estrés, pero le está costando mucho trabajo acomodar ese tiempo con su trabajo y su familia ya que su doctor actual se encuentra muy lejos y le toma bastante tiempo el ir. Por esto decidí buscar en la aplicación de telemedicina que le recomendó un amigo si ofrecen esta especialidad para poder tomar la cita desde su teléfono o tableta en cualquier momento sin necesidad de desplazarse.
objetivo	Validar si en la aplicación existe la especialidad médica que necesita

Funcionalidades	<ul style="list-style-type: none"> - Inicio de sesión. - Visualizar la lista de Especialidades que se tienen en la aplicación.
Desarrollo de tareas	<ul style="list-style-type: none"> - [Registro] - Inicio de sesión - Abrir listado de Especialidades disponibles

Escenario 4 Programar una cita con un médico especialista	
Perfil de usuario	Paciente
Personaje	Pedro Gutiérrez
Contexto	Pedro quiere programar una cita con su psiquiatra para continuar con su terapia contra el estrés, ahora que ha encontrado la aplicación de telemedicina y ya se encuentra registrado quiere probarla.
objetivo	Programar una cita para una fecha determinada en la especialidad requerida
Funcionalidades	<ul style="list-style-type: none"> - Listado de Especialidades disponibles. - Programar una cita con un especialista
Desarrollo de tareas	<ul style="list-style-type: none"> - registro - Inicio de sesión - Buscar en lista de Especialidades la que se requiere. - Programar una cita.

Escenario 5 Tomar una cita programada con el médico especialista	
Perfil de usuario	Paciente
Personaje	Pedro Gutiérrez
contexto	Pedro ha programado una cita con un psiquiatra para el día de hoy a las 18 horas, la va a tomar desde su oficina al terminar el horario laboral y así probar seguir con su terapia en esta modalidad.
objetivo	Tener una video consulta a partir de la cita programada previamente.
Funcionalidades	<ul style="list-style-type: none"> - Ver cita programada - Video consulta con el medico
Desarrollo de tareas	<ul style="list-style-type: none"> - Registro - Inicio de sesión - Ver cita programada - Video consulta con el medico

Escenario 6 Tomar una cita no programada con un medico	
Perfil de usuario	Paciente
Personaje	Carla Pérez
contexto	Carla se siente un mal debido a un resfriado el día de hoy y quiere hablar con un médico sobre el tratamiento que puede tomar para disminuir los síntomas, por lo que opta por tener una consulta con un médico mediante la aplicación de telemedicina.
objetivo	Tener una cita no programada en la aplicación.
Funcionalidades	- Vídeo consulta con un medico
Desarrollo de tareas	- Registro - Inicio de sesión - Vídeo consulta con el medico

Escenario 7 Responder a una cita no programada	
Perfil de usuario	Médico
Personaje	Carlos Gonzales
contexto	Carlos ha terminado su turno en su Trabajo de la mañana y ahora va a comenzar a trabajar utilizando la aplicación de telemedicina con la cual atiende a los pacientes los cuales tienen dificultad para ir hasta su consultorio.
objetivo	Tomar una consulta de un paciente que se encuentra en espera
Funcionalidades	- Ver pacientes en espera - Vídeo conferencia
Desarrollo de tareas	- Registro - Inicio de sesión - Ver pacientes en espera - Video consulta con el primer paciente

Escenario 8 Responder a una cita programada	
Perfil de usuario	Médico
Personaje	Carlos Gonzales
contexto	Carlos ha terminado su turno en su Trabajo de la mañana y ahora va a comenzar a trabajar utilizando la aplicación de telemedicina con la cual atiende a los pacientes los cuales tienen dificultad para ir hasta su consultorio.

objetivo	Tomar una consulta programada previamente por un paciente
Funcionalidades	<ul style="list-style-type: none"> - Ver citas programadas - Video conferencia
Desarrollo de tareas	<ul style="list-style-type: none"> - Registro - Inicio de sesión - Ver citas programadas - Tomar la cita programada para la hora especificada.

Escenario 9 Registro de paciente por comando de voz	
Perfil de usuario	Paciente
Personaje	Pedro Pérez
contexto	Debido a su enfermedad Pedro tiene bastantes citas médicas, pero él quiere comenzar a tomarlas por su cuenta desde casa por lo que busca una aplicación de telemedicina que le permita usarla mediante comandos de voz
objetivo	Registrarse en la aplicación mediante comandos de voz
Funcionalidades	<ul style="list-style-type: none"> - Registro de un paciente
Desarrollo de tareas	<ul style="list-style-type: none"> - Buscar una aplicación en la Play Store. - Descargar la aplicación. - Registrarse en la aplicación colocando sus datos personales.

Escenario 10 Programar una cita con medico mediante comandos de voz	
Perfil de usuario	Paciente
Personaje	Pedro Pérez
contexto	Pedro quiere programar una cita con unos de los especialistas que lo ayudan con su enfermedad.
objetivo	Programar una cita con un especialista
Funcionalidades	<ul style="list-style-type: none"> - Listado de Especialidades disponibles. Programar una cita con un especialista
Desarrollo de tareas	<ul style="list-style-type: none"> - Registro - Inicio de sesión - Buscar en lista de Especialidades la que se requiere. Programar una cita.

Escenario 11 tomar una cita programada mediante comandos de voz	
Perfil de usuario	Paciente
Personaje	Pedro Pérez
contexto	Pedro quiere tomar una cita con un especialista a las 17:00 horas que ya había programado previamente y es una de las consultas que tiene de forma regular.
objetivo	Tomar un vídeo consulta de una cita con un especialista que fue programada previamente.
Funcionalidades	<ul style="list-style-type: none"> - Ver cita programada - Video consulta con el medico
Desarrollo de tareas	<ul style="list-style-type: none"> - Registro - Inicio de sesión - Ver cita programada - Video consulta con el medico

2.1.2.3 Flujos de interacción

A continuación, se presentan los flujos de interacción de los usuarios con las aplicaciones móviles. Estos permiten conocer de qué manera el usuario puede interactuar con las distintas funcionalidades de la aplicación, ya que muestran de una forma gráfica todas las decisiones y posibles caminos que estos pueden tomar dentro de la aplicación. También nos permiten encontrar requerimientos funcionales o de información en cualquier punto de los flujos.

En la siguiente imagen encontramos el diagrama del flujo de interacción del usuario con la aplicación. En la ilustración 1 se tiene el diagrama para los pacientes y en la ilustración 2 se tiene el diagrama para los médicos.

El primer diagrama de flujos de interacción es el del paciente, en el cual se puede ver el como el usuario puede utilizar la aplicación desde que la abre hasta que logre algunos de los objetivos que se encuentran planteados en los escenarios de uso. Al abrir la aplicación el paciente debe registrarse para poder comenzar a utilizar el servicio, luego puede iniciar sesión. Una vez iniciada la sesión, el paciente tiene acceso a las funcionalidades principales de la aplicación, las cuales se encuentran ahí para satisfacer las necesidades encontradas durante las fases anteriores.

La funcionalidad primordial es la video conferencia con el médico, la cual como se puede ver en el flujo de interacción, se puede iniciar desde el home si el paciente quiere tomar una cita en el momento, en otro caso puede programar una cita que puede tomar en otro momento o tomar una cita que ya ha programado previamente.

El flujo muestra que para tener acceso a la aplicación siempre se debe iniciar sesión, esto es debido a que se van a manejar datos personales y médicos, a

los cuales nadie debe tener acceso y tampoco deben ser divulgados de ninguna manera, por lo tanto, se mantiene ese nivel de seguridad.

El segundo diagrama muestra el flujo de interacción del doctor con la aplicación, en este caso el medico también debe registrarse para poder utilizar los servicios de la aplicación, y debe iniciar sesión siempre para poder tener acceso a las funcionalidades.

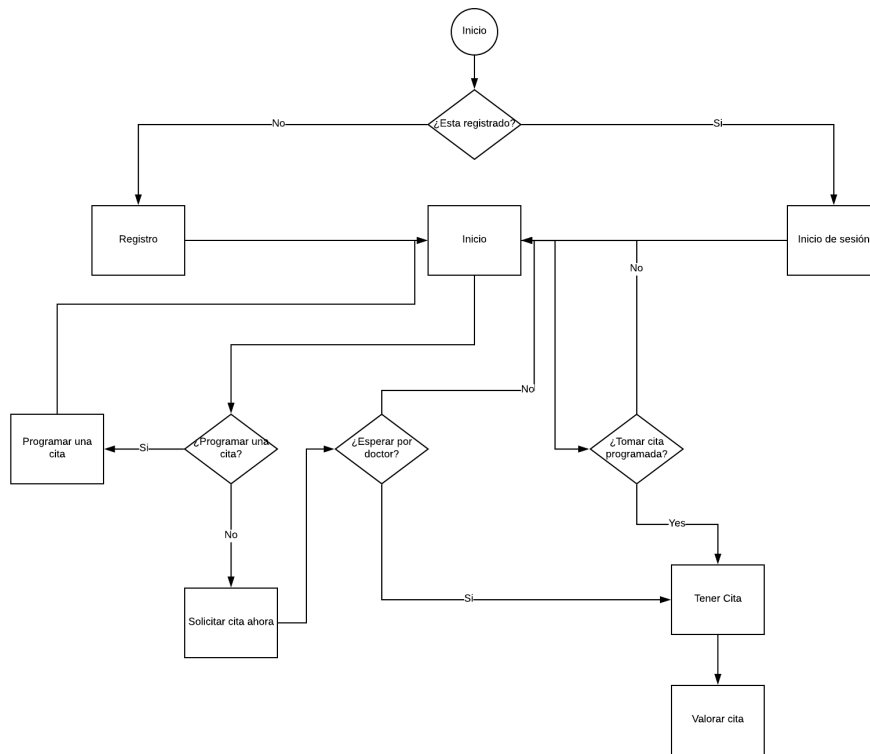


Ilustración 4. Diagrama de flujo para pacientes

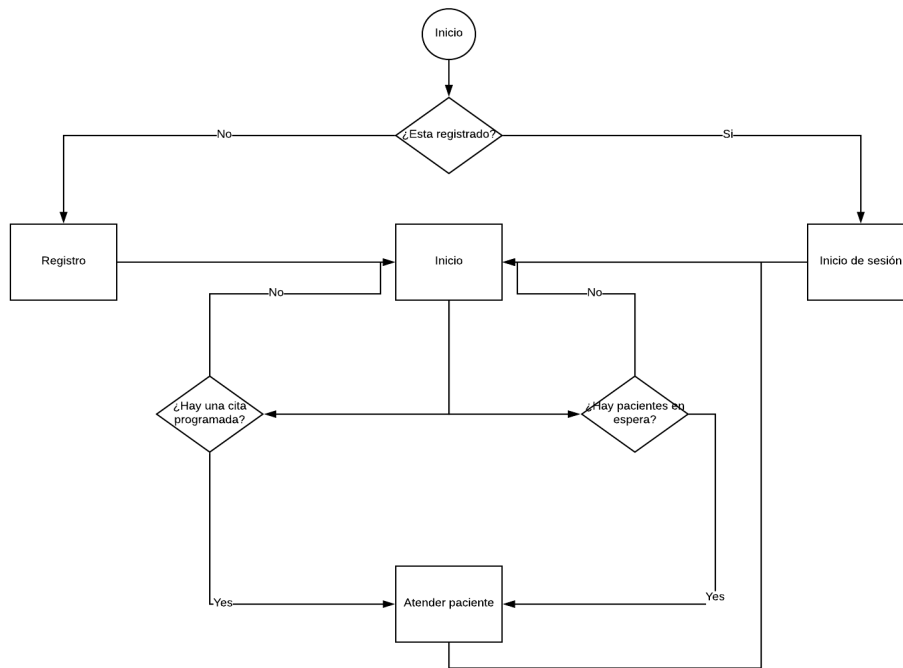
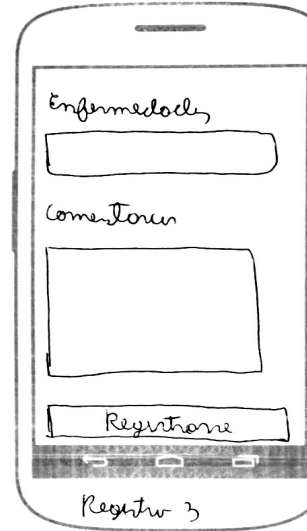
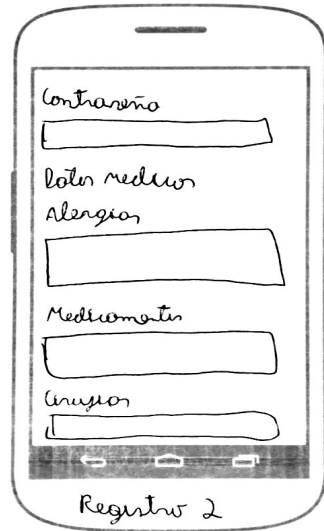
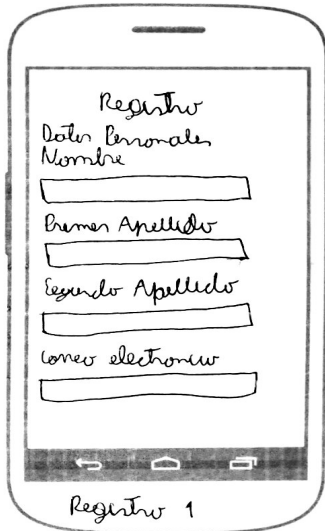
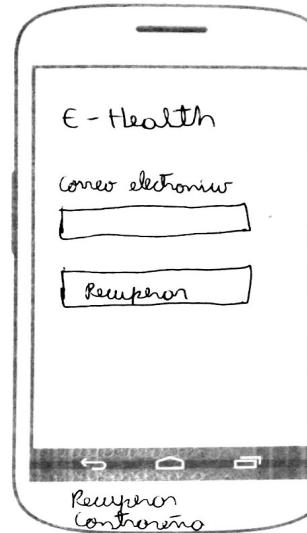
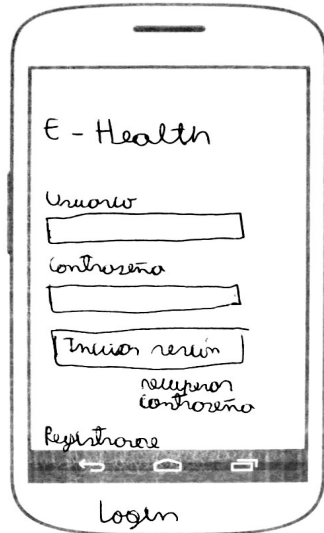
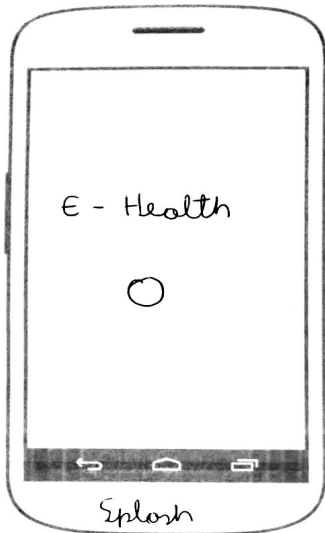


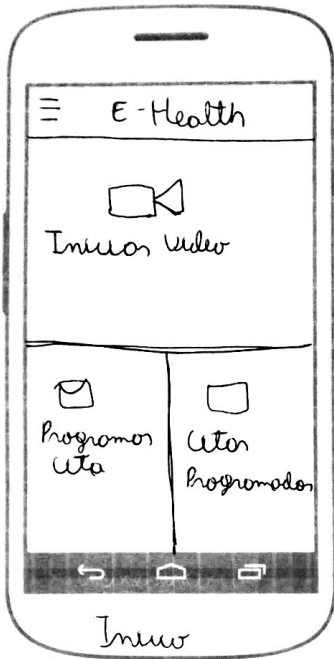
Ilustración 5. Diagrama de flujo para doctores

2.1.3 Prototipado

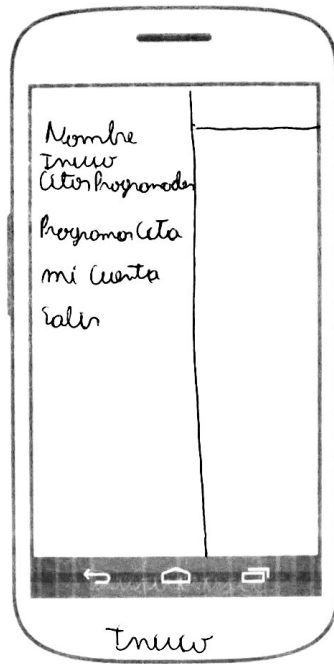
En esta fase del diseño centrado en el usuario se realiza el prototipo, el cual consiste en la elaboración uno o varios modelos del sistema que se diseña. Este proceso se divide en dos partes, en la primera se crean los prototipos de baja fidelidad los cuales se utilizan para evaluar ideas de diseño, distribución de los elementos en la interfaz gráfica y su prioridad. Además, se utilizan como base para los prototipos de alta fidelidad.

Los prototipos de alta fidelidad se utilizan para obtener un modelo más cercano a la versión final del producto en la cual se pueden evaluar de una forma más precisa los elementos de diseño y usabilidad del producto.

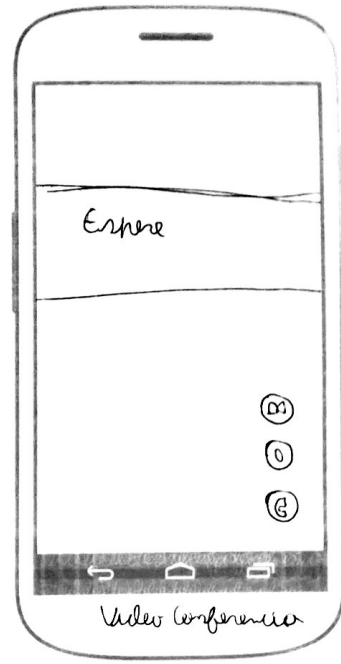




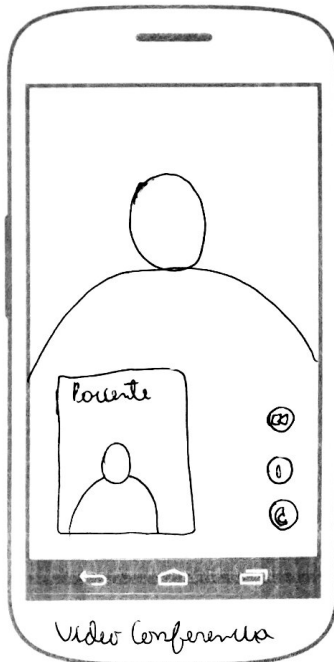
Inicio



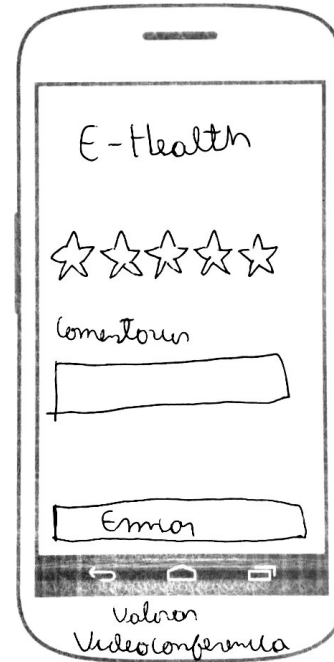
Inicio



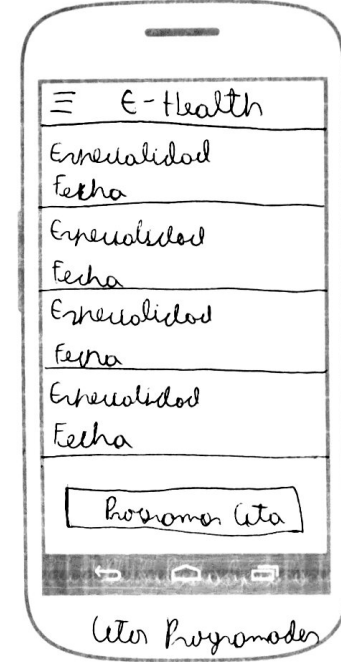
Video Conferencia



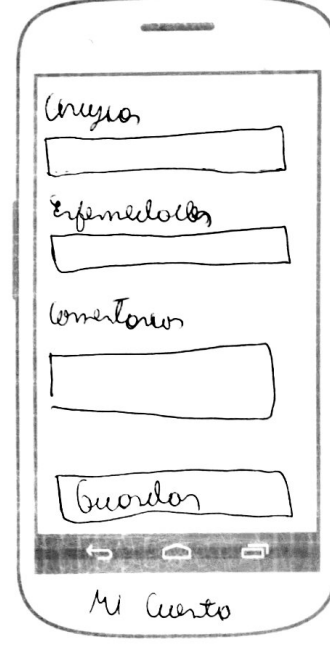
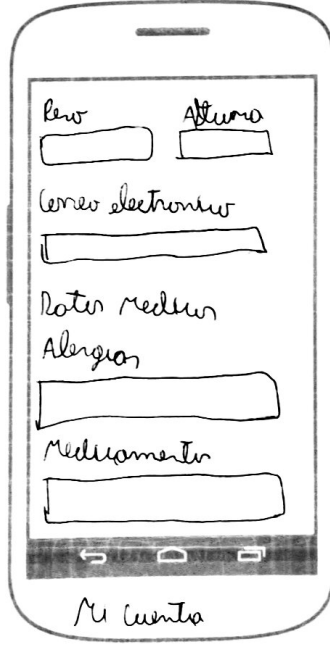
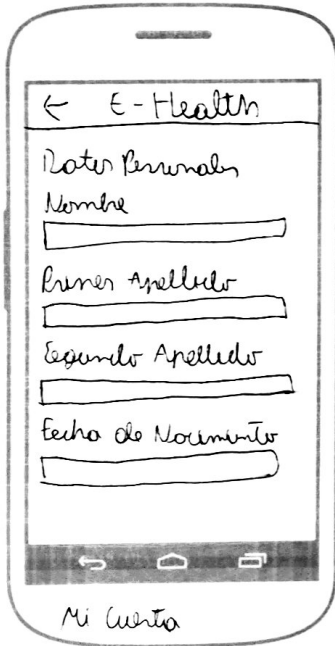
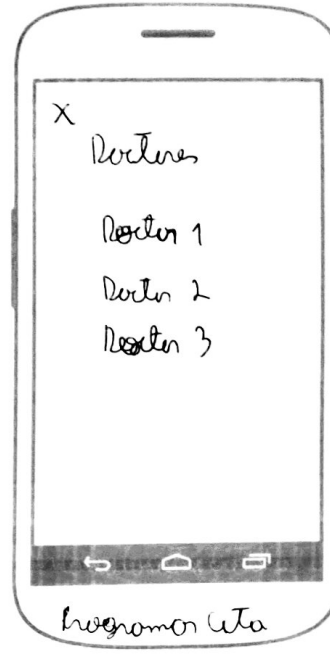
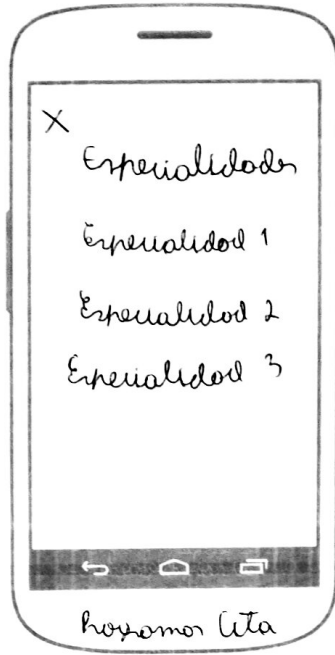
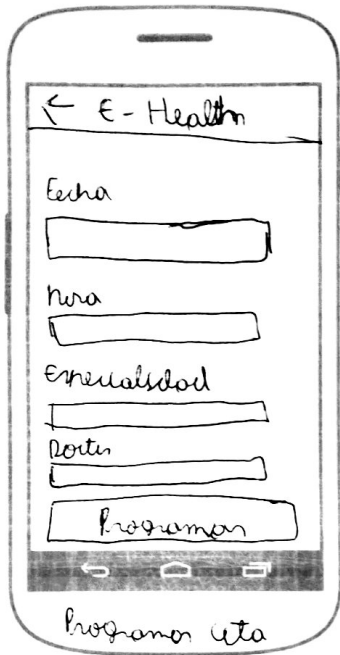
Video Conferencia

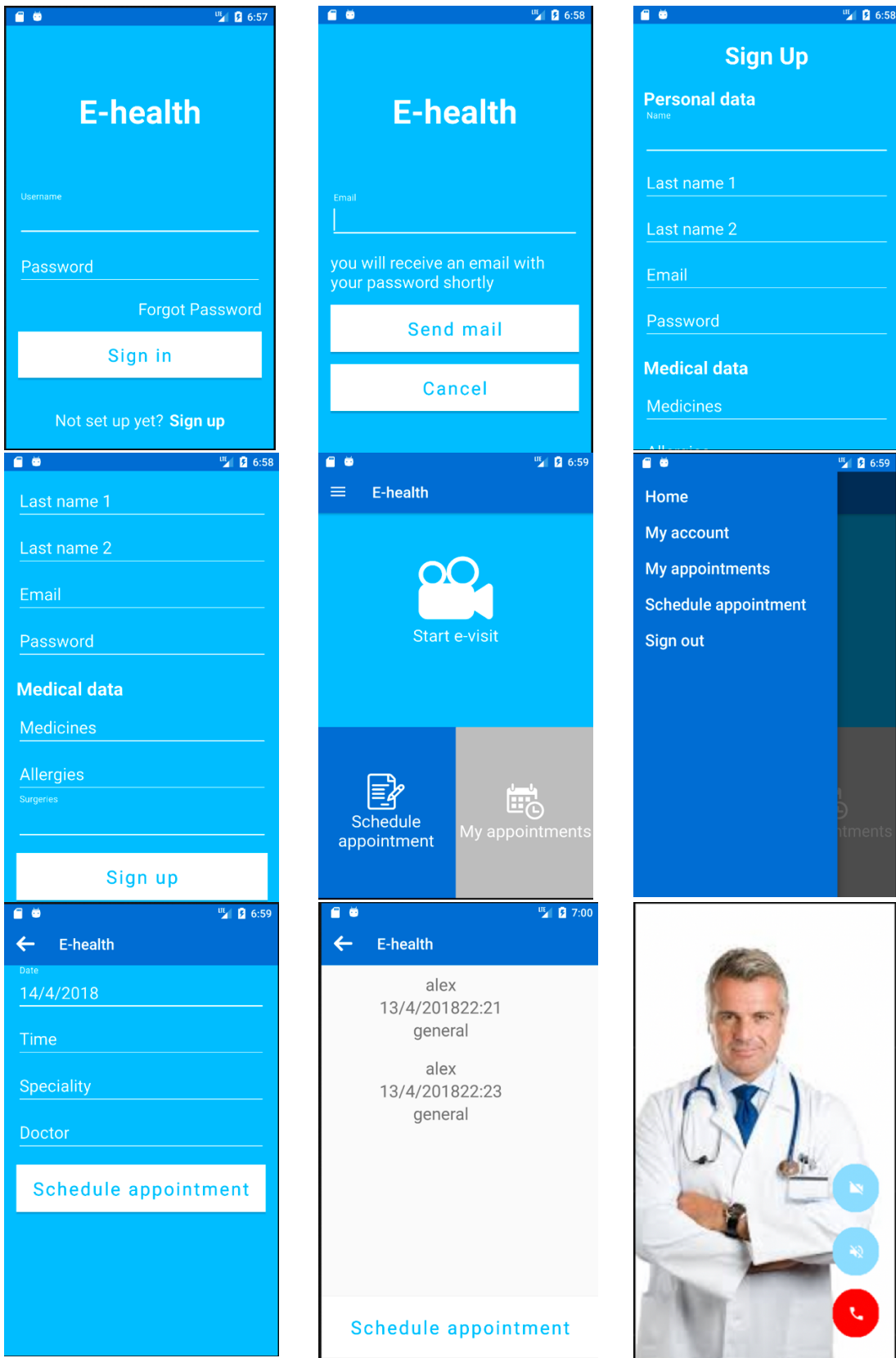


Valorar Videoconferencia



Uta Programador

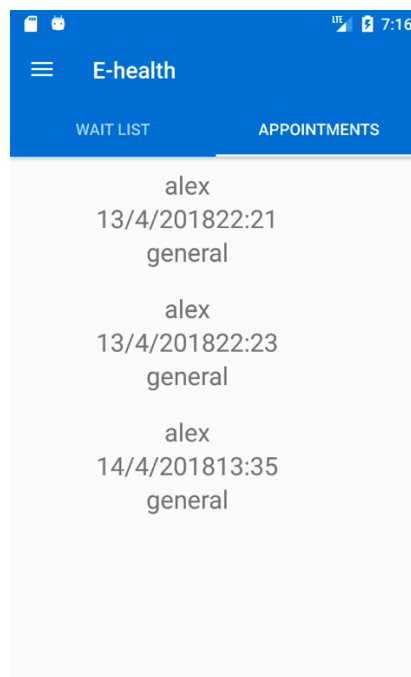
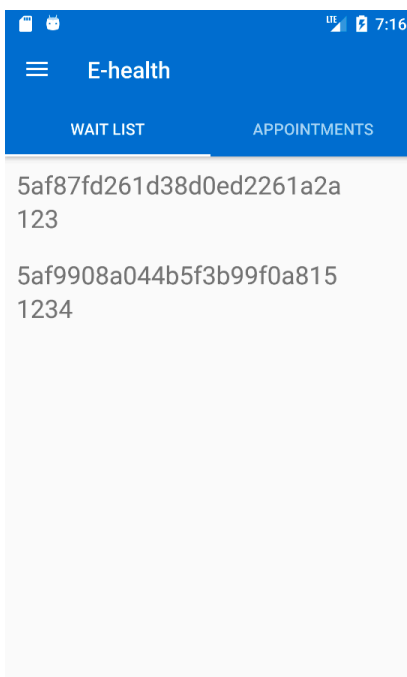




A continuación, una breve explicación de cada una de las pantallas del prototipo del paciente:

- 1- Splash: esta pantalla es la primera que se muestra al abrir la aplicación, en este caso solo se encarga de mostrar el nombre de E-Health y redirigir al usuario a la pantalla de login.
- 2- Inicio de sesión: esta pantalla permite tener acceso a 3 funcionalidades de la aplicación, recuperar contraseña, registrarse e iniciar sesión para poder tener acceso a la aplicación.
- 3- Recuperar contraseña: esta pantalla solicita el email del usuario para que este pueda recuperar su contraseña.
- 4- Registro: esta pantalla le solicita al usuario la información necesaria para crear una cuenta de acceso a la aplicación, se solicitan tanto datos personales básicos como médicos.
- 5- Home: esta pantalla se muestra luego de iniciar sesión en la aplicación y permite tener acceso a las funcionalidades principales de la aplicación como la video conferencia, programar una cita, o ver las citas programadas previamente, además contiene un menú lateral que también da acceso a las opciones de mi cuenta, y cerrar sesión.
- 6- Video conferencia: esta pantalla es en la cual el paciente habla con el médico, puede llegar aquí desde una cita programada o por demanda.
- 7- Programar una cita: esta pantalla permite programar una cita con medico en un día y hora específico.
- 8- Lista de citas programadas: esta pantalla muestra las citas que han sido programadas previamente por el paciente.
- 9- Mi cuenta: en esta pantalla el paciente puede actualizar sus datos personales y médicos, además puede completar la información solicitada por la aplicación.
- 10- Evaluar (Rate): esta pantalla se encuentra luego de la video conferencia y permite colocar una valoración sobre la atención recibida.

Prototipo de alta definición de la aplicación que utilizara el medico:



En el prototipo de la aplicación que va a utilizar el médico no se muestran todas las pantallas ya que comparte varias con el diseño del paciente y se van a describir únicamente las que tienen diferencias en diseño y funcionalidad.

- 1- Pantalla de registro: en esta pantalla se piden datos personales del médico y especialidad además de los datos necesarios para crear la cuenta en la aplicación.
- 2- Home: esta pantalla contiene un Tab menú con dos secciones, una en la cual se muestran a los pacientes en espera de ser atendidos y las citas programadas.
- 3- Mi cuenta: esta pantalla, al igual que en el caso del paciente se utiliza para que el médico complete y actualice la información de su cuenta.

Para complementar los prototipos mostrados anteriormente y mejorar la accesibilidad de la aplicación para personas con visión o movilidad reducida se plante utilizar las opciones de accesibilidad que el sistema operativo Android provee, especialmente dos de sus características como son el talk back y voice actions.

Talk back: es una característica de accesibilidad del sistema operativo Android con la cual el dispositivo lee todo lo que se encuentra en la pantalla del dispositivo y las opciones seleccionadas, esto permite que, aunque el paciente no distinga por completo las opciones, si pueda escuchar las distintas opciones que se le presentan.

Voice actions: es una característica de accesibilidad del sistema operativo Android que una vez activada permite seleccionar mediante comandos de voz cualquier elemento de la pantalla, lo que permite el uso de cualquier aplicación sin necesidad de tocar el dispositivo.

2.1.4 Evaluación

Debido a la naturaleza de la aplicación que se busca desarrollar en este trabajo, realizar una prueba de usabilidad del prototipo con usuarios reales es muy complicado debido a la logística de encontrar dichos usuarios y que puedan realizar las pruebas necesarias. Estos usuarios idealmente serían personas con movilidad o visión reducida, personas que puedan utilizar este servicio de telemedicina de forma regular y médicos que puedan utilizar la aplicación para contestar las vídeo llamadas.

Se descarta la idea de utilizar cualquier otro tipo de usuario para la evaluación del prototipo ya que eso daría resultados poco fiables, los cuales darían lugar a conclusiones erróneas sobre la usabilidad de la aplicación.

Se realizará una evaluación heurística como la propone Nielsen, la cual indica los siguientes puntos a evaluar:

Visibilidad del estado del sistema: El sistema siempre debe indicar lo que está sucediendo, debe proveer retroalimentación al usuario en un tiempo razonable.

Coherencia entre el sistema y el mundo real: El sistema debe comunicarse con el usuario con palabras, frases y términos que este pueda entender.

Control y libertad del usuario: en muchos casos los usuarios van a seleccionar funcionalidades por error y el sistema los debe proveer de forma visible y simple una forma de salir de dicha funcionalidad.

Consistencia y estándares: Los usuarios no deben preguntarse si las palabras, situaciones o acciones significan lo mismo. Se deben utilizar estándares de la plataforma.

Prevención de errores: se debe ser cuidadoso al diseñar el sistema para evitar ser propenso a errores.

Reconocer y no recordar: se debe minimizar lo que el usuario debe recordar para poder utilizar el sistema, las instrucciones de uso siempre deben ser visibles y simples.

Flexibilidad y eficiencia de uso: tener aceleradores o atajos para que los usuarios experimentados puedan tener una interacción más veloz

Diseño estético y minimalista: Los diálogos no deben contener información irrelevante.

Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores: los mensajes de error deben ser expresado en un lenguaje sencillo, deben ser precisos al indicar el error y sugerir una solución.

Ayuda y documentación: aunque es mejor si el sistema puede ser utilizado sin documentación, puede ser necesario tener ayuda y documentación.

Los puntos anteriormente mencionados, se utilizaron durante la fase de diseño de los prototipos generando varias iteraciones sobre los mismos hasta que se consideraron suficientemente Buenos como para ser utilizados en la aplicación.

2.2 Diseño técnico

Utilizando toda la información obtenida del apartado de diseño centrado en el usuario será utilizada para crear los casos de uso, los módulos del sistema y una arquitectura que se ajuste a las necesidades del sistema.

2.2.1 Casos de uso

En este apartado se van a definir los casos de uso del proyecto. En primer lugar, vamos a definir a los actores

Actor-001	Usuario
Descripción	Este actor representa a todos los usuarios definidos con el perfil de

paciente que interactúen con la aplicación en un dispositivo móvil

Actor-002	Usuario
Descripción	Este actor representa a todos los usuarios definidos con el perfil de médico que interactúen con la aplicación en un dispositivo móvil

A continuación, en la ilustración 3 se muestra el diagrama con los paquetes de casos de uso del sistema.

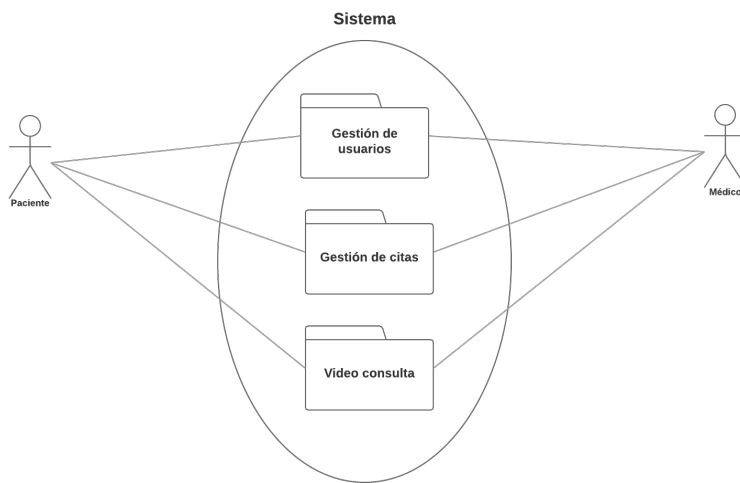


Ilustración 6. Paquetes de casos de uso

En la ilustración 4 se muestran los detalles del paquete de gestión de usuarios.

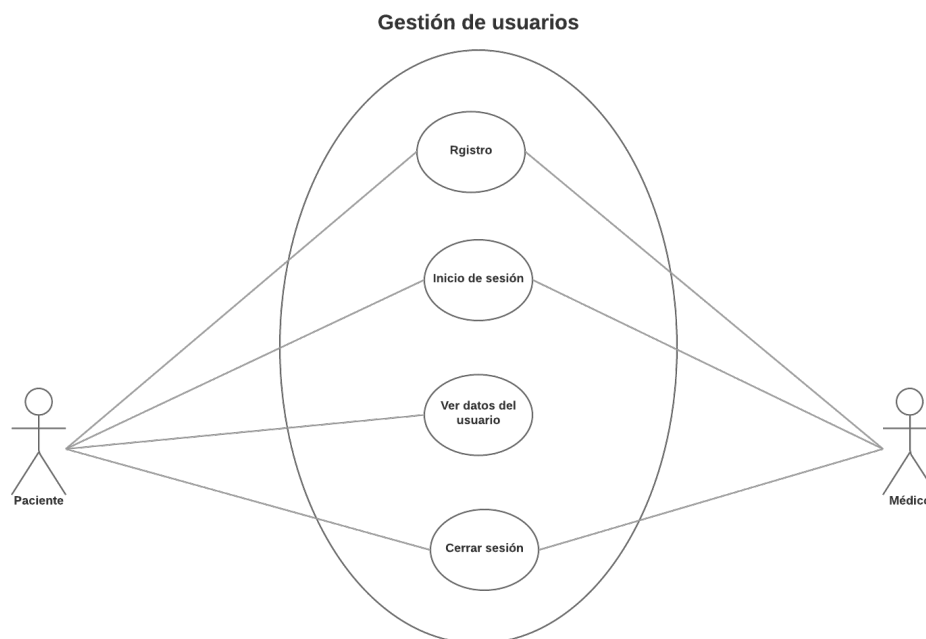


Ilustración 7. Gestión de usuarios.

En la ilustración 5 se muestran los detalles del paquete de gestión de citas.

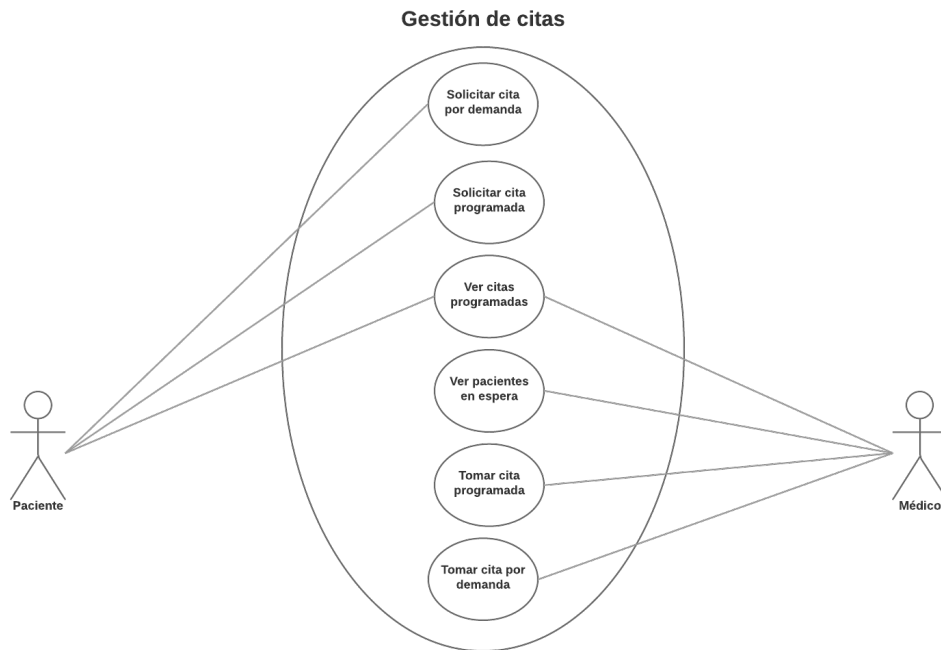


Ilustración 8. Gestión de citas

En la ilustración 6 se muestran los detalles del paquete de vídeo consulta.

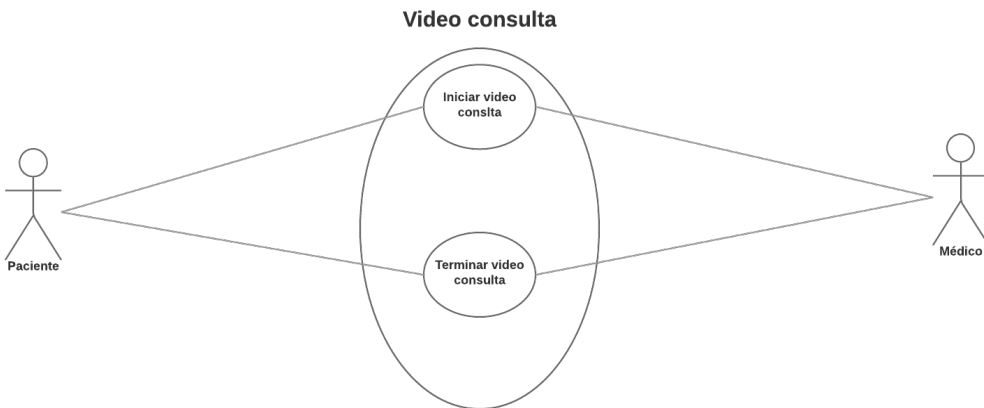


Ilustración 9. Video consulta

A continuación, se muestran en detalle los casos de uso mencionados en los diagramas anteriores

CU-001	Registro paciente
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios registrarse dentro de la aplicación
Actores	Paciente
Precondiciones	No estar registrado en la aplicación

Iniciado por	Paciente
Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Acceder al registro - Rellenar el formulario con nombre, apellidos, correo electrónico, contraseña, alergias, medicamentos que toma actualmente, cirugías que ha sufrido, enfermedades que tenga actualmente, cualquier otro comentario que crea relevante sobre su historia medica - Pulsar el botón de registro - El sistema valida todos los campos y que el usuario no registrara previamente - El sistema crea el usuario y lo redirige a la pantalla de inicio
Flujo de excepción	<ul style="list-style-type: none"> - Si todos los campos obligatorios no están llenos se mostrará un mensaje de error al usuario. - Si el usuario ya está registrado, se le mostrara un mensaje de error
Post condiciones	El sistema registrara al usuario
notas	Al registrarse correctamente, también se realiza el login del usuario

CU-002	Iniciar sesión
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios registrados tener acceso a la aplicación
Actores	Paciente, medico
Precondiciones	<ul style="list-style-type: none"> - Realizar el caso CU-001 para el paciente - Realizar el caso CU-004 para el medico - Recordar usuario y contraseña
Iniciado por	Paciente, medico
Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Colocar usuario y contraseña - Presionar el botón de inicio de sesión - Validar credenciales del usuario - Si la validación es exitosa, se

	dirigí al usuario a la pantalla de inicio
Flujo de excepción	- Si la validación no es correcta se muestra un mensaje de error al usuario
Post condiciones	- El usuario inicia sesión en la aplicación
notas	

CU-003	Cerrar sesión
Prioridad	Normal
Descripción	Este caso de uso permite que el usuario cierre una sesión activa de la aplicación
Actores	Paciente, medico
Precondiciones	- Haber realizado el caso de uso CU-002
Iniciado por	Paciente, medico
Flujo	<ul style="list-style-type: none"> - Presionar el botón de cerrar sesión en el menú lateral. - Cerrar sesión el usuario - Dirigir al usuario a la pantalla de login.
Flujo de excepción	
Post condiciones	Se cerrará la sesión activa del usuario en la aplicación
notas	

CU-004	Registro de medico
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios registrarse dentro de la aplicación
Actores	Medico
Precondiciones	No estar registrado en la aplicación
Iniciado por	Medico
Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Acceder al registro - Rellenar el formulario con nombre, apellidos, correo electrónico, contraseña, especialidad - Pulsar el botón de registro - El sistema valida todos los campos y que le usuario no registrara previamente - El sistema crea el usuario y lo redirige a la pantalla de inicio

Flujo de excepción	<ul style="list-style-type: none"> - Si todos los campos obligatorios no están llenos se mostrará un mensaje de error al usuario. - Si el usuario ya está registrado, se le mostrara un mensaje de error
Post condiciones	El sistema registrara al usuario
notas	Al registrarse correctamente, también se realiza el login del usuario

CU-005	Solicitar Cita médica por demanda
Prioridad	Alta
Descripción	Este caso de uso permite a los usuarios solicitar una consulta luego de iniciar sesión en la aplicación
Actores	paciente
Precondiciones	<ul style="list-style-type: none"> - Haber realizado el caso de uso CU-002
Iniciado por	paciente
Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Iniciar sesión - Presionar la opción de solicitar cita médica en este momento - Esperar a ser atendido.
Flujo de excepción	
Post condiciones	El paciente solicita la consulta médica y se coloca en la sala de espera
notas	

CU-006	Solicitar una cita programada
Prioridad	Alta
Descripción	Este caso de uso permite solicitar una cita programa con un especialista
Actores	Paciente
Precondiciones	Haber realizado el caso de uso CU-002
Iniciado por	Paciente
Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Iniciar sesión - Presionar la opción de programar cita desde el menú lateral o desde la pantalla de inicio. - Llenar el formulario con la fecha de la cita, la hora de la cita, la especialidad médica y

	<p>el doctor.</p> <ul style="list-style-type: none"> - Presionar el botón de programar
Flujo de excepción	<ul style="list-style-type: none"> - Si los campos obligatorios del formulario no están llenos se le mostrará un mensaje de error al usuario
Post condiciones	El usuario solicita una cita programada
notas	

CU-007	Recuperar contraseña
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios recuperar su contraseña en caso de haberla olvidado
Actores	Paciente, medico
Precondiciones	<ul style="list-style-type: none"> - Haber realizado el caso de uso CU-001 para el caso de paciente. - Haber realizado el caso de uso CU-004 para el caso de medico
Iniciado por	Paciente, medico
Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Presionar el botón de recuperar contraseña. - Llenar el formulario con el correo electrónico. - Presionar el botón de recuperar contraseña
Flujo de excepción	<ul style="list-style-type: none"> - Si el campo del correo está vacío se le mostrara un mensaje de error al usuario. - Si la cuenta de correo no está registrada se le mostrara un mensaje de error al usuario
Post condiciones	Se le envía un correo la contraseña al usuario
notas	

CU-008	Tener un vídeo consulta por demanda
Prioridad	Alta
Descripción	Este caso de uso les permite a los usuarios tener una vídeo consulta por demanda con un medico
Actores	paciente
Precondiciones	Haber realizado el caso de uso CU-

	005
Iniciado por	Paciente
Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Iniciar sesión - Presionar la opción de solicitar cita médica en este momento - Esperar a ser atendido. - Tener la consulta médica.
Flujo de excepción	
Post condiciones	El paciente tiene una consulta médica por demanda
notas	

CU-009	Tener una vídeo consulta de una cita programada
Prioridad	Alta
Descripción	Este caso de uso les permite a los usuarios tener una vídeo consulta a partir de una cita médica programada con un especialista
Actores	Paciente
Precondiciones	Haber realizado el caso de uso CU-006
Iniciado por	Paciente
Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Iniciar sesión - Ver sus citas programadas. - Presionar el botón de iniciar de la cita correspondiente - Esperar a ser atendido - Tener la consulta <p>Presionar el botón de programar</p>
Flujo de excepción	
Post condiciones	El paciente tiene una vídeo consulta a partir de una cita programada con un especialista
notas	

CU-010	Valorar una vídeo consulta
Prioridad	Baja
Descripción	Este caso de uso permite a los usuarios valorar la calidad de la consulta médica
Actores	Paciente
Precondiciones	Haber realizado el caso de uso CU-008 o CU-009
Iniciado por	Paciente
Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Iniciar sesión

	<ul style="list-style-type: none"> - Presionar la opción de solicitar cita médica en este momento Ver sus citas programadas. - Presionar el botón de iniciar de la cita correspondiente - Esperar a ser atendido. - Tener la consulta médica.
Flujo de excepción	
Post condiciones	El usuario valora la calidad de la consulta medica
notas	

CU-011	Ver datos de la cuenta
Prioridad	Baja
Descripción	Este caso de uso permite a los usuarios ver sus datos personales y médicos desde la aplicación.
Actores	Paciente
Precondiciones	Haber realizado el caso de uso CU-002
Iniciado por	Paciente
Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Iniciar sesión - Desplegar el menú lateral - Presionar la opción de mi cuenta - Ver datos de la cuenta
Flujo de excepción	<ul style="list-style-type: none"> - Si existe algún error de comunicación con el servidor para obtener los datos se le mostrara un mensaje de error al usuario
Post condiciones	El paciente ve todos sus datos ingresados durante el proceso de registro.
notas	

CU-012	Modificar datos de la cuenta
Prioridad	baja
Descripción	Este caso de uso le permite a un usuario modificar tanto sus datos personales como sus datos médicos dentro de la aplicación.
Actores	Paciente
Precondiciones	Haber realizado el caso de uso CU-011
Iniciado por	Paciente
Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Iniciar sesión

	<ul style="list-style-type: none"> - Desplegar el menú lateral - Presionar la opción de mi cuenta - Ver datos de la cuenta - Llenar campos vacíos del formulario - Modificar los campos que se quieran modificar - Presionar el botón de guardar
Flujo de excepción	<ul style="list-style-type: none"> - Si alguno de los campos obligatorios se encuentra vacío se le mostrará un mensaje de error al usuario
Post condiciones	El usuario completa sus datos personales y médicos en la aplicación
notas	

CU-013	Ver pacientes en espera de cita por demanda
Prioridad	Alta
Descripción	Este caso de uso permite que un usuario pueda ver a los pacientes que se encuentran en espera a ser atendidos.
Actores	Medico
Precondiciones	Haber realizado el caso de uso CU-002
Iniciado por	Medico
Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Iniciar sesión - Presionar en el tab de pacientes en espera - Ver lista de pacientes en espera
Flujo de excepción	
Post condiciones	El usuario puede ver la lista de los pacientes en espera a ser atendidos
notas	

CU-014	Tomar una cita por demanda
Prioridad	Alta
Descripción	Este caso de uso les permite a los usuarios poder seleccionar e iniciar una cita por demanda
Actores	Medico
Precondiciones	Haber realizado el caso de uso CU-013
Iniciado por	Medico

Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Iniciar sesión - Presionar en el tab de pacientes en espera - Ver lista de pacientes en espera - Seleccionar al primer paciente de la lista - Presionar el botón de iniciar llamada
Flujo de excepción	
Post condiciones	El medico inicia la vídeo llamada con un paciente
notas	

CU-015	Ver citas programadas
Prioridad	Alta
Descripción	Este caso de uso permite que los usuarios puedan ver la lista de citas programadas que tienen asignadas.
Actores	Medico
Precondiciones	Haber realizado el caso de uso CU-002
Iniciado por	Medico
Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Iniciar sesión - Presionar en el tab de citas programadas - Ver lista de citas programadas
Flujo de excepción	
Post condiciones	El medico ve el listado de citas programadas
notas	

CU-016	Tomar una cita programada
Prioridad	Alta
Descripción	Este caso de uso permite a los usuarios iniciar una cita programada.
Actores	Medico
Precondiciones	Haber realizado el caso de uso CU-015
Iniciado por	Medico
Flujo	<ul style="list-style-type: none"> - Abrir la aplicación - Iniciar sesión - Presionar en el tab de citas programadas - Ver lista de citas programadas

	<ul style="list-style-type: none"> - Seleccionar la primera cita programada - Presionar el botón de iniciar llamada
Flujo de excepción	
Post condiciones	El medico iniciar la vídeo llamada para la cita programada
notas	

2.2.2 Modelo de datos

En esta sección se indica el diseño del modelo de datos que se utilizará en el servidor para almacenar toda la información del sistema.

Ya que se plantea utilizar como manejador de base de datos mongoDB y esta es una base de datos no SQL, se colocan los modelos de clases para representar los objetos json almacenados.

E-HEALTH MODELO DE DATOS

Alexander Cabezas | May 14, 2018

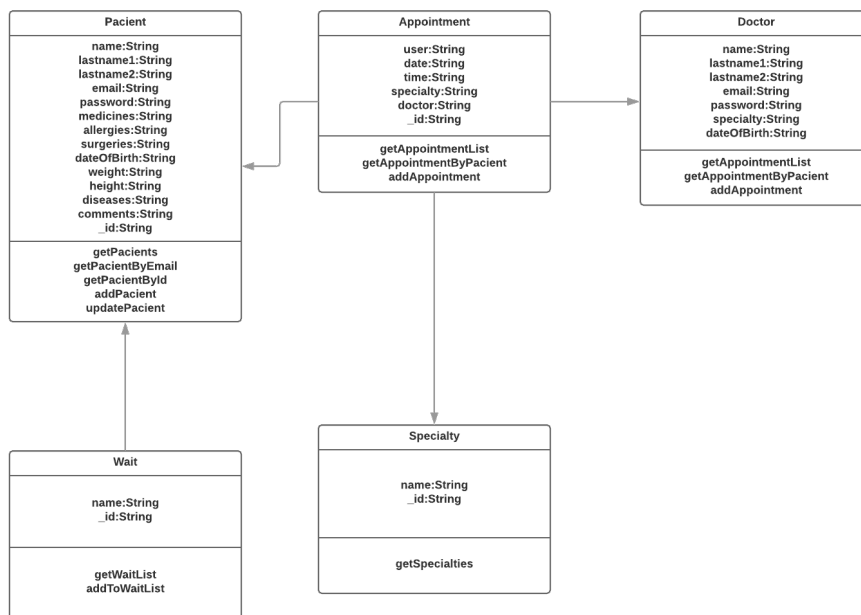


Ilustración 10. Modelo de datos

Ahora se explicará cada elemento del diagrama de datos:

- 1- Wait: contiene a los pacientes en la lista de espera para ser atendidos por médico.
- 2- Specialty: contiene a las especialidades médicas disponibles en la aplicación.

- 3- Doctor: contiene toda la información de los doctores registrados en la aplicación.
- 4- Pacient: contiene toda la información de los pacientes registrados en la aplicación.
- 5- Appointment: contiene todas las citas programadas por los pacientes en la aplicación.

2.2.3 Diseño de la arquitectura

En este apartado se transformarán las ideas y requerimientos obtenidos en los apartados anteriores en modelos más cercanos a la solución que se quiere obtener. El objetivo de este apartado es obtener la estructura de los distintos módulos del sistema y la interacción entre ellos.

El sistema que se pretende desarrollar utilizara una arquitectura cliente-servidor, este es un modelo en el cual la aplicación se distribuye entre los proveedores de servicios o recursos a los cuales les denominaremos servidores y los demandantes, a los cuales denominaremos clientes. Los clientes realizan peticiones a los servidores y estos les envían una respuesta.

En este tipo de modelo la capacidad de procesamiento necesario para llevar a cabo las tareas del sistema es divididas entre los clientes y los servidores. Sin embargo, lo más importante que ofrece este modelo es la centralización en los servidores de la gestión de la información y también la separación de responsabilidades entre los clientes y los servidores.

En el caso particular de este proyecto, los clientes serán las aplicaciones para dispositivos móviles Android, las cuales serán las encargadas de realizar las peticiones de información al servidor para poder ser mostrada a los usuarios a través de la interfaz gráfica.

El servidor que se plantea ser desarrollado en nodejs se encargaran de responder las peticiones de las aplicaciones (clientes) para autenticar a los usuarios, obtener información etc.

2.2.3.1 Arquitectura de la aplicación

Para el desarrollo de la aplicación android se utilizará el patrón de arquitectura VIPER que por sus siglas en ingles significa View Interactor Presenter Entity Router. Este patrón de arquitectura es una evolución del patrón MVP el cual por sus siglas en ingles es Model View Presenter.

El patrón de arquitectura VIPER se basa en los principios SOLID al separar en capas y clases particulares las responsabilidades y así hacer que la aplicación sea más flexible y fácil de mantener.

Cada uno de los componentes de la aplicación se va clasificar dentro de los elementos definidos dentro del patrón:

View (Vista): Este tipo de componente contiene los elementos visuales, no contiene ningún tipo de lógica de negocio y se separa por completo de los otros componentes para hacer posible la modificación de la interfaz gráfica sin necesidad de modificar ningún otro componente.

Presenter (Presentador): este tipo de componente funciona como puente entre la vista y el interactor, además de conocer los flujos de la aplicación dependiendo de los resultados que se obtengan del interactor o de las acciones que se lleven a cabo en la vista.

Router (Enrutador): este tipo de componente contiene todos los posibles flujos que tienen desde una actividad en el caso de Android.

Entity (Entidad): este tipo de componente se refiere a los modelos de datos que se van a utilizar en la aplicación, son el equivalente a los POJO de java.

Interactor: este tipo de componente es el que contiene la lógica de negocio.

2.2.3.2 Vista física

Por último, en esta sección se habla sobre la estructura física del sistema. Esta vista pretende mostrar cómo va estar distribuido el sistema entre los distintos componentes de hardware.

Lo principal en esta vista es mostrar cómo se encuentra creada la arquitectura cliente servidor. En la ilustración 7 se presenta el diagrama correspondiente.

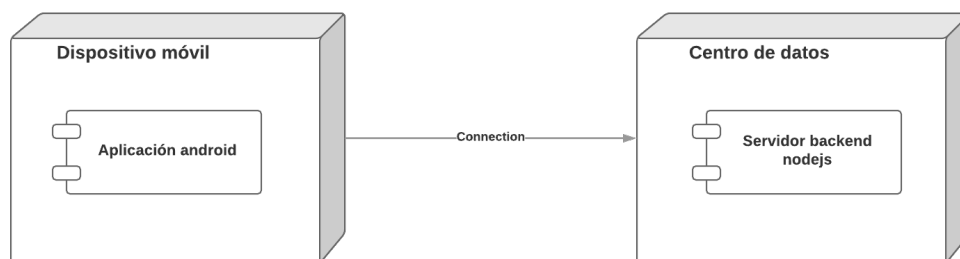


Ilustración 11. Vista física de las aplicaciones

3. Implementación

3.1 Herramientas y APIS utilizadas

Durante esta fase se han utilizado múltiples herramientas tanto de software como de hardware para poder llevar a cabo el desarrollo de este sistema y que además ayudaron a facilitar dicho proceso al ofrecer soluciones ya culminadas y probadas. En la siguiente imagen se muestra un conjunto de las herramientas utilizadas y luego se explicará en profundidad el uso de algunas de ellas.



3.1.1 Android

Android es un sistema operativo basado en linux para dispositivos móviles con pantalla táctil, tabletas, relojes inteligentes, televisores y automóviles, es desarrollado y mantenido por Google.

3.1.2 AndroidStudio

AndroidStudio es un IDE (Integrated Development Environment) especialmente diseñado para el desarrollo de aplicaciones Android, soporta los lenguajes oficiales de desarrollo de la plataforma como Java y Kotlin. Esta herramienta es desarrollada y mantenida por JetBrains.

3.1.3 Gradle

Gradle es una herramienta de software libre que permite automatizar el proceso de compilación de una aplicación, se caracteriza por tener un muy

buen desempeño y ser muy flexible, permite la ejecución de scripts escritos en los lenguajes de programación groovy y kotlin.

3.1.4 NodeJS

NodeJs es un entorno de ejecución para el lenguaje de programación JavaScript que se encuentra construido sobre el motor JavaScript V8 de Google Chrome. Este entorno usa un modelo de entrada y salida sin bloqueo y orientado a eventos lo que lo hace notablemente ligero y eficiente.

3.1.5 MongoDB

MongoDB es un sistema de manejo de bases de datos noSQL orientado a documentos y es un software libre. Esta herramienta forma parte de una nueva familia de manejadores de bases de datos noSQL que no utilizan tablas como los manejadores de bases de datos clásicos, si no que se encuentran basados en documentos, que son estructuras flexibles similares a un JSON con esquemas dinámicos.

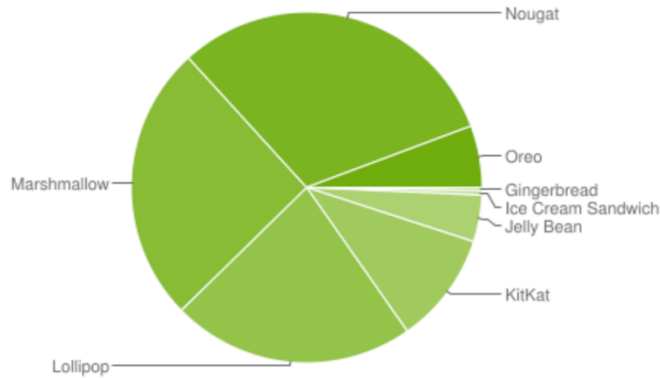
3.2 Aspectos relevantes del desarrollo

3.2.1 Proyecto Android

Como ya se ha mencionado anteriormente este sistema cuenta con dos aplicaciones móviles, una para el uso del paciente y otra para el uso del doctor. Ambas aplicaciones se encuentran en proyectos distintos, la decisión de crear dos proyectos separados y no solamente uno en el cual las diferencias se desarrollasen mediante Android product flavors es que las aplicaciones comparten solamente los estilos, la funcionalidad para iniciar la video conferencia y estructura de las pantallas, sin embargo, utilizan endpoints diferentes, objetos de dominio distintos, tienen lógica de negocio y flujos de interacción distintos.

Por lo mencionado anteriormente se consideró que agregar la complejidad necesaria para manejar android product flavors para el desarrollo de ambas aplicaciones no sopesaba el duplicar las únicas partes similares entre las aplicaciones.

Por otro lado, ambas aplicaciones fueron desarrolladas de forma totalmente nativa en AndroidStudio y son compatibles hasta la versión 5.0 Lollipop de Android (API 21), lo que representa el 88.8 % de todos los dispositivos Android registrados en la Play Store. Se decidió no soportar versiones más antiguas del sistema operativo Android para dar mejor soporte a las nuevas actualizaciones y características de dispositivos modernos las cuales existen en versiones tan antiguas del sistema operativo. A continuación, se muestra una gráfica obtenida sobre la cantidad de dispositivos móviles distribuidos por versiones del sistema operativo.



Data collected during a 7-day period ending on May 7, 2018.
 Any versions with less than 0.1% distribution are not shown.

Ilustración 12. Distribución de versiones de android

Como se mencionó en la fase de diseño, el patrón de arquitectura elegido para el desarrollo de ambas aplicaciones es VIPER, el cual permite una separación de responsabilidades mediante capas en la aplicación, en las siguientes figuras se muestran ejemplos de la implementación de este patrón de arquitectura en algunas funcionalidades. Además, cada aplicación cuenta con un módulo especializado en manejar las peticiones a red, el cual fue desarrollado con una combinación de los patrones de desarrollo repositorio y fabrica, lo que permite separar cada fuente de datos, como la local y la remota u otras si existiera el caso. Se muestra una imagen con la estructura de este módulo para la aplicación del paciente.

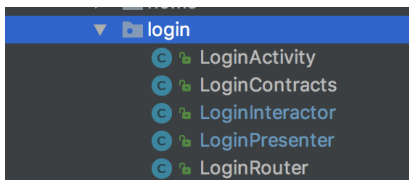


Ilustración 13. Estructura de paquete login

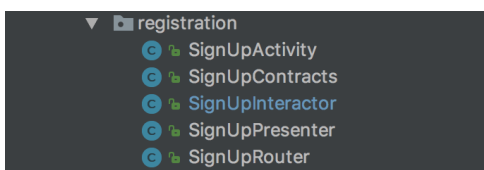


Ilustración 14. Estructura de paquete para el registro

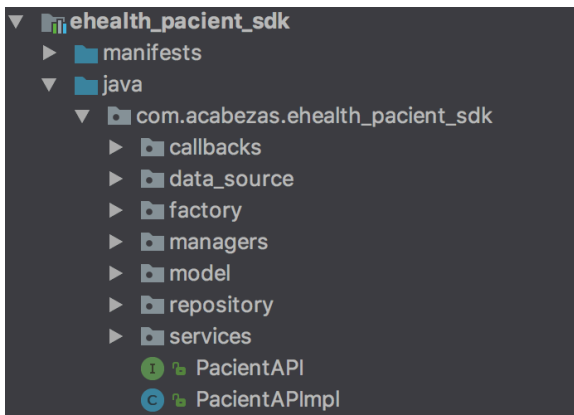


Ilustración 15. Estructura del sdk del paciente

3.2.1.1 Implementación de interfaces de usuario

Debo a la gran fragmentación de dispositivos Android en cuanto tamaños y densidades de pantallas y con el propósito de ofrecer un producto que se pueda utilizar en la mayor cantidad de dispositivos posibles, todas las pantallas de la aplicación fueron desarrolladas de una forma flexible para que se adaptaran a las distintas densidades de pantallas disponibles en el mercado, tomando en cuenta las mejores prácticas de desarrollo de interfaces propuestas por google.

Entre algunas de las consideraciones tomadas durante el proceso de desarrollo se destaca el hecho de que los layouts de las pantallas fueron diseñados de forma flexible, de tal manera que se ajustan a la pantalla de acuerdo a su densidad, esto se logró utilizando el `ConstraintLayout` de Android el cual permite colocar los componentes visuales de forma relativa entre sí, y con respecto a los bordes de la pantalla, esto permite que dichos elementos mantenga su posición relativa sin importar el tipo de pantalla.

Además, para establecer el largo y alto de los componentes visuales se utilizaron los valores de `fill_parent` y `wrap_content` en los atributos que establecen dichas características en los componentes, esto ayuda a mantener un layout flexible, también los tamaños fijos de textos o altura y largo vienen dados en unidades independientes a los pixeles como son `sp` para el caso de tamaños de textos y `dp` para el caso de tamaño de imágenes y otros componentes visuales.

3.2.3 Servidor

Otra de las piezas fundamentales del sistema es el backend, pues sin este las aplicaciones móviles no podrían ofrecer ninguna de las funcionalidades. Esta aplicación almacenada en un servidor se utiliza para conectarse a la base de datos y obtener la información requerida, además de ofrecer a las aplicaciones móviles un api de consulta donde estas pueden solicitar la información que necesite el usuario en un momento dado o según la funcionalidad que se encuentre utilizando.

Para el proceso de desarrollo que instaló una base de datos y servidor local sobre el siguiente equipo de trabajo, MacBook Pro de 13 pulgadas, con procesador Intel core i5 de 2,3 GHZ, 8 GB de memoria ram y 128 GB de SSD.

Como se mencionó en apartados anteriores el manejador de bases de datos seleccionado fue mongoDB, por lo que este fue instalado de forma local en el equipo, la versión de mongoBD utilizada fue la 3.6.

También fue instalado de forma local nodeJS para el desarrollo de la aplicación de backend, la versión de nodeJs utilizada fue la 8.11.1, además en el proyecto se utilizaron componentes para facilitar el proceso de desarrollo como fueron:

- Express: este componente o librería, es un framework que permite crear una infraestructura de forma rápida y eficiente para nodeJS, que proporciona características para el desarrollo de APIS para aplicaciones móviles y web.
- Body-Parser: este componente traduce el cuerpo de una petición de red en un middleware antes que los manejadores, lo que nos da acceso directo a un objeto.
- Mongoose: este complemento permite modelar los datos almacenados en mongoDB en forma de esquemas, también incluye casting de tipos, validaciones y métodos para crear queries.

A continuación, se presentan los esquemas utilizados para crear los modelos con mongoose para poder manejar los datos almacenados en mongoDB.

```
var waitSchema = mongoose.Schema({
  name: {
    type:String
  }
});
```

Ilustración 16. Modelo de lista de espera

```
var specialtySchema = mongoose.Schema({
  name: {
    type:String
  }
});
```

Ilustración 17. Modelo de especialidades

```

▼ var appointmentSchema = mongoose.Schema({
  ▼ user: {
    type:String
  },
  ▼ date: {
    type:String
  },
  ▼ time: {
    type:String
  },
  ▼ specialty: {
    type:String
  },
  ▼ doctor: {
    type:String
  }
});

```

Ilustración 18. Modelo de citas

```

var doctorSchema = mongoose.Schema({
  name: {
    type:String
  },
  lastname1: {
    type:String
  },
  lastname2: {
    type:String
  },
  email: {
    type:String
  },
  password: {
    type:String
  },
  specialty: {
    type:String
  },
  dateOfBirth: {
    type:String
  }
});

```

Ilustración 19. Modelo de doctor

```

var pacienteSchema = mongoose.Schema({
  name: {
    type:String
  },
  lastname1: {
    type:String
  },
  lastname2: {
    type:String
  },
  email: {
    type:String
  },
  password: {
    type:String
  },
  medicines: {
    type:String
  },
  allergies: {
    type:String
  },
  surgeries: {
    type:String
  },
  dateOfBirth: {
    type:String
  },
  weight: {
    type:String
  },
  height: {
    type:String
  },
  diseases: {
    type:String
  },
  comments: {
    type:String
  }
});

```

Ilustración 20. Modelo de paciente

Para poder tener acceso a los esquemas se desarrolló un API por medio de express el cual, al ser consumido por las aplicaciones móviles puedan obtener los datos requeridos para su funcionamiento. A continuación, se muestran imágenes con los endpoints desarrollados.

```

/*
DOCTOR SIGNUP endpoint
*/
app.post('/api/doctor/signup',function(req,res){ ... });

/*
DOCTOR LOGIN endpoint
*/
app.post('/api/doctor/login',function(req,res){ ... });

/*
DOCTOR GET DATA endpoint
*/
app.get('/api/:id/doctor_profile_data',function(req,res){ ... });

/*
DOCTOR UPDATE DATA endpoint
*/
app.put('/api/:id/doctor/update_doctor_profile_data',function(req,res){ ... });

/*
DOCTOR RECOVER PASSWORD endpoint
*/
app.post('/api/doctor/recover_password',function(req,res){ ... });

/*
GET PACIENT IN WAIT LIST endpoint
*/
app.get('/api/doctor/get_user_waitlist',function(req,res){ ... });

/*
GET SCHEDULED APPOINTMENTS
*/
app.get('/api/doctor/:id/scheduled_appointments',function(req,res){ ... });

```

Ilustración 21. Endpoints de doctores.

```

/*
*****
START PACIENT ENDPOINTS
*****
*/
/*
GET DOCTORS endpoint
*/
▶ app.get('/api/doctors',function(req,res){ ... });

/*
GET SPECIALTIES endpoint
*/
▶ app.get('/api/specialties',function(req,res){ ... });

/*
PACIENT SIGNUP endpoint
*/
▶ app.post('/api/signup',function(req,res){ ... });

/*
PACIENT LOGIN endpoint
*/
▶ app.post('/api/login',function(req,res){ ... });

/*
PACIENT RECOVER PASSWORD ENDPOINT
*/
▶ app.post('/api/recover_password',function(req,res){ ... });

/*
GET PACIENT DATA ENDPOINT
*/
▶ app.get('/api/:id/user_profile_data',function(req,res){ ... });

/*
UPDATE PACIENT DATA ENDPOINT
*/
▶ app.put('/api/:id/user_profile',function(req,res){ ... });

/*
ADD PACIENT TO WAITLIST
*/
▶ app.post('/api/add_user_waitlist',function(req,res){ ... });

/*
SCHEDULE APPOINTMENT endpoint
*/
▶ app.post('/api/schedule_appointment',function(req,res){ ... });

```

Ilustración 22. Endpoints de pacientes

3.3 Pruebas

Al culminar el proceso de desarrollo de las aplicaciones móviles y de la aplicación del backend se realizaron pruebas de aceptación de las diferentes funcionalidades de la aplicación. Estas pruebas fueron realizadas en emuladores del sistema operativo con las siguientes características:

- Nexus 5 con sistema operativo Android 6.0 API 23.

- 1.5 GB de memoria ram.
- VM heap de 128 MB
- 800 MB de almacenaje interno
- CPU de dos núcleos con arquitectura x86

Las funcionalidades que se validaron de forma manual fueron:

Paciente:

- Login.
- Recuperar contraseña.
- Registro.
- Programar cita.
- Listar citas programadas.
- Video conferencia.
- Agregar paciente a la lista de espera.

Doctor

- Login.
- Recuperar contraseña.
- Registro.
- Listar citas programadas.
- Listar pacientes en espera.
- Video conferencia.

3.4 Conclusiones de la implementación

El proceso de desarrollo fue llevado con éxito tras múltiples iteraciones sobre las distintas aplicaciones, en cada iteración se pudieron observar errores y mejoras que fueron implementadas en las siguientes. Estos cambios introdujeron modificaciones sobre los diseños de fases anteriores los cuales también fueron actualizados.

4. Conclusiones

Al culminar el trabajo de fin de master se tienen algunas conclusiones que se presentan a continuación.

En el ámbito personal, este trabajo ha contribuido a aumentar y afianzar los conocimientos adquiridos durante el master y mi carrera profesional. En particular he podido consolidar mis conocimientos en el desarrollo de aplicaciones para dispositivos con sistema operativo Android, así como también el desarrollo de un backend y de todo el ciclo de desarrollo de una aplicación móvil.

El desarrollo de este proyecto ha sido un enorme reto personal al cual le he invertido bastante tiempo, y. creo que cumple satisfactoriamente con los objetivos planteados, obteniendo un producto funcional. Por otro lado, en la memoria de este trabajo he intentado explicar de la mejor manera posible todos los procesos llevados a cabo durante el desarrollo de este sistema y justificar las decisiones tomadas.

La tarea más compleja fue el desarrollo del backend ya que es donde tengo menos experiencia, así como el diseño e implementación de la base de datos noSql. Estas tareas fueron en las que se invirtió más tiempo durante la implementación del sistema.

Si se toman en cuenta los objetivos planteados al inicio de este trabajo, se puede concluir que todos han sido cumplidos en su totalidad. El sistema cumple con todos los requerimientos funcionales y de diseño expuestos en los apartados del inicio de este trabajo.

En cuanto a la gestión del proyecto, es importante denotar que una buena planificación es necesaria para llevar un proyecto a buen término, esto incluye tiempos de diseño y desarrollo realistas tomando en cuenta la cantidad de recursos disponibles y fechas de entrega.

En el caso de este proyecto se intentó seguir la planeación lo más posible lo cual ha permitido cumplir con las entregas en las fechas planteadas. Sin embargo, durante la fase de implementación se tuvo una desviación sobre la planificación, ya que el backend tuvo retrasos debido a la falta de conocimientos sobre las tecnologías utilizadas, lo cual causó también retraso en la planificación del proceso de desarrollo de las aplicaciones móviles, esto fue subsanado al trabajar más horas de las previstas en el proyecto.

En cuanto a la metodología utilizada para llevar la planificación del proceso, considero que utilizar scrum fue un éxito como ejercicio académico ya que me permitió entender mejor su funcionamiento y como puede ser aplicada a un proyecto de desarrollo de software no solo a la fase de implementación, si no a todas las fases dentro del ciclo de vida de la elaboración del sistema.

Para culminar, me gustaría agregar que este proceso de desarrollo ha sido extraordinario en muchos ámbitos y que me ha permitido llevar mis

conocimientos y capacidades a un nivel, el cual utilizare en mi vida tanto profesional como académica.

5. Glosario

Aplicación. Se refiere un o una parte de un sistema informático.

TFM. Trabajo final de master.

Android. Sistema operativo basado en Linux para dispositivos móviles.

AndroidStudio. Ambiente de desarrollo de aplicaciones para Android.

IDE. Integrated development environment o ambiente de desarrollo integrado. Es una herramienta para el desarrollo de aplicaciones que provee funcionalidades para facilitar dicho proceso.

API. Application Programming Interface o Interfaz de programación de aplicaciones. Es un conjunto de métodos o funciones que ofrece una cierta librería o biblioteca para ser utilizado por otro software.

DCU. Diseño centrado en el usuario.

Backend. Aplicación que se encuentra desplegada en un servidor que es consumido por una aplicación móvil y devuelve la información correspondiente.

VIPER. View Interactor Presenter Entity Router. Patrón de diseño de arquitectura.

DBMS. Data base management system.

APK. Android application package es un paquete para instalar aplicaciones en el dispositivo con sistema operativo Android.

JetBrains. Empresa encargada del desarrollo y mantenimiento del IDE AndroidStudio.

JSON. Javascript object notation, es un formato ligero para el intercambio de datos.

6. Bibliografía

Molich, R. and Nielsen, J., Improving a human- computer dialogue, Communications of the ACM, 33(3), 338-348, (1990).

Nielsen, J., Enhancing the explanatory power of usability heuristics, CHI'94 Conference Proceedings, (1994).

<http://www.usabilitynet.org/> 21/03/2018

<https://docs.gradle.org/current/userguide/userguide.html> 15/05/2018

<https://es.wikipedia.org/wiki/MongoDB> 15/05/2018

<https://developer.android.com/about/dashboards/> 15/05/2018

<https://developer.android.com/studio/intro/?hl=es-419> 15/05/2018

7. Anexos

- Manual de usuario correspondiente a la aplicación del paciente. Este manual de usuario contiene toda la información necesaria para que un paciente pueda conocer y utilizar todas las funcionalidades de la aplicación.
- Manual de usuario correspondiente a la aplicación del médico. Este manual de usuario contiene toda la información necesaria para que un doctor pueda conocer y utilizar todas las funcionalidades de la aplicación.
- Manual de instalación y configuración. En este manual se explica cómo se deben instalar y configurar los componentes necesarios para la compilación y ejecución del proyecto.
- Código fuente. Todo el código fuente correspondiente a las dos aplicaciones móviles y la aplicación del backend se encuentra en archivos comprimidos .zip adjuntos a este trabajo.