

AGENDA PERSONAL MÉDICA PARA DISPOSITIVOS MÓVILES

**GERALDINE CAICEDO ORTIZ
LAURA MILENA LEÓN MÉNDEZ**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2017**

AGENDA PERSONAL MÉDICA PARA DISPOSITIVOS MÓVILES

**GERALDINE CAICEDO ORTIZ
LAURA MILENA LEÓN MÉNDEZ**

**Trabajo de grado para optar al título de Ingeniería
de Sistemas**

**Director
PEDRO JAVIER TRUJILLO TARAZONA
Ing. de Sistemas, Mg. Informática**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2017**

DEDICATORIA

A Dios por esta gran oportunidad que me otorgó, con cada día de vida, con la que me bendijo y los cuales aproveche para cumplir con mis metas.

A mis padres Betty y Rodrigo por siempre estar ahí apoyándome especialmente en los momentos más difíciles en los cuales quería renunciar.

A mis maestros durante esta carrera que no solo me enseñaron la teoría sino como aplicarla creando grandes bases para mi futuro

A Milena por luchar a mi lado para conseguir este fin mutuo.

Muchas gracias.

Geraldine Caicedo Ortiz

Primero Dios por darme la sabiduría no solo suficiente sino con añadidura en cada decisión tomada y darme oportunidad de vivir y culminar con éxito este logro propuesto.

Mis padres Nabor y Emilse por su apoyo y fé en mí además de ser aquellas personas que me dieron un excelente ejemplo y construyeron unos pilares fuertes que me acompañaran toda la vida.

Mis hermanos y sobrinos por ese amor impartido que me alegraban cada momento.

Mis amigos de la Universidad cómplices de locuras en esta aventura que inicio años atrás y compañeros a lo largo de este camino que nos llevó a esta meta.

A Geraldine mi compinche en esta última etapa presentada aquí.

A cada profesor que estuvo a lo largo que mi vida no solo formando una profesional sino una persona, especialmente a ese profesor que con sus consejos me alegraba las tardes.

A la Universidad por abrirme la puerta a mi segundo hogar y formarme profesionalmente.

A todos de corazón Gracias.

Milena León Méndez

TABLA DE CONTENIDO

INTRODUCCIÓN	18
1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA	20
2. OBJETIVOS.....	22
2.1. OBJETIVO GENERAL	22
2.2. OBJETIVOS ESPECÍFICOS	22
3. MARCO TEORICO	23
3.1. SALUD	23
3.1.1. Medicamentos	23
3.1.2. Plan Obligatorio de Salud (POS)	24
3.1.3. Medicamentos POS	24
3.1.4. Medicamentos NO POS	24
3.1.5. Consultas Médicas	25
3.1.6. Exámenes Médicos	25
3.2. TECNOLOGÍAS INVOLUCRADAS EN EL DESARROLLO DEL PROYECTO	25
3.2.1. JavaScript	25
3.2.2. JQuery	26
3.2.3. Hypertext Pre-Processor (PHP)	26
3.2.4. Asynchronous JavaScript y XML (AJAX)	26
3.2.5. Lenguaje de Marcado de Hypertexto (HTML)	27
3.2.6. Hojas de Estilo en Cascada (Cascading Style Sheets (CSS))	28
3.2.7. MySQL	28
3.2.8. Common Gateway Interface (CGI)	29
3.2.9. PhoneGap	29
3.2.10. Apache Cordova	30
3.2.11. Modelo Vista Controlador (MVC)	37
4. METODOLOGÍA	38
4.1. FASES DE LA EJECUCIÓN	39
4.1.1. Fase de Inicio	39
4.1.2. Fase de Planeación	39
4.1.3. Fase de Construcción	39
4.1.4. Fase de Pruebas	39
5. FASE DE PLANEACIÓN.....	40
5.1. REQUISITOS FUNCIONALES	40
5.2. REQUISITOS NO FUNCIONALES	41

5.3.	CASOS DE USOS	41
5.3.1.	Identificación de Actores	41
5.3.2.	Descripción Diagrama Modelo Casos de Usos:	43
5.3.3.	Descripción del Diagrama de Caso de Uso – Gestión Información por Usuario	45
5.3.4.	Diagrama de Caso de Uso – Gestión Información Usuarios	47
6.	FASE DE CONSTRUCCIÓN.....	50
6.1.	BASE DE DATOS	50
6.2.	MOCKUPS DE LA APLICACIÓN	51
6.3.	IMPLEMENTACIÓN	55
6.3.1.	Componente Modelo	56
6.3.2.	Componente Vista	57
6.3.3.	Componente Controlador	58
6.3.4.	APIs Usados.	59
6.4.	RESULTADOS DE LA IMPLEMENTACIÓN DEL PROTOTIPO	60
6.4.1.	Versión Móvil	60
6.4.2.	Versión Web.	61
6.4.3.	Implementación Base de Datos	61
6.4.4.	Implementación Apk	62
6.5.	ARQUITECTURA	63
7.	FASE DE PRUEBAS.....	65
7.1.	PRUEBAS DE USABILIDAD	65
7.2.	PASOS QUE SE SIGUIERÓN	65
8.	CONCLUSIONES	66
9.	RECOMENDACIONES	67
10.	BIBLIOGRAFÍA	68
11.	ANEXOS.....	70

LISTA DE FIGURAS

Figura 1 Diferencia de petición-respuesta del modo tradicional vs al usar Ajax.....	27
Figura 2 Esquema de Ejecución CGI.....	29
Figura 3 Identificador de Actores	41
Figura 4 Diagrama Modelos de Casos de Usos.....	43
Figura 5 Diagrama Gestión de Información	45
Figura 6 Diagrama Gestión de Usuarios	47
Figura 7 Modelo Entidad Relación de la Base de Datos	45
Figura 8 Mockup Página de Inicio	51
Figura 9 Mockup Formulario de Registro	51
Figura 10 Mockup Inicio de Sesión	52
Figura 11 Mockup Menú Principal	52
Figura 12 Mockup Módulo Citas	53
Figura 13 Mockup Módulo Medicamentos	53
Figura 14 Mockup Módulo Médicos	54
Figura 15 Mockup Módulo Exámenes	54
Figura 16 Mapa mental sobre el uso de la aplicación	55
Figura 17 Ejemplo código para un modelo	56
Figura 18 Ejemplo del código vista	57
Figura 19 Ejemplo código de un controlador	58
Figura 20 Código para acceder a la API de envió de notificaciones push	59
Figura 21 Pantallazo de la aplicación móvil (Módulo medicamentos).....	60
Figura 22 Pantallazo Instalación del Apk	62

Figura 23 Arquitectura del despliegue de la página web y de la aplicación móvil....63

Figura 24 Diagrama de la arquitectura la página web y de la aplicación móvil.....64

LISTA DE TABLAS

Tabla 1 Descripción del actor administrador	42
Tabla 2 Descripción del actor administrador	42
Tabla 3 Descripción Caso de Uso Autenticación	43
Tabla 4 Descripción Caso de Uso Gestionar Información	44
Tabla 5 Descripción Caso de Uso Gestionar Usuarios	44
Tabla 6 Descripción Caso de Uso Calificación	45
Tabla 7 Descripción Caso de Uso Consultar	46
Tabla 8 Descripción Caso de Uso Editar (Usuario)	46
Tabla 9 Descripción Caso de Uso Eliminar (Usuario)	47
Tabla 10 Descripción Caso de Uso Consultar	48
Tabla 11 Descripción Caso de Uso Editar (Administrador)	48
Tabla 12 Descripción Caso de Uso Eliminar (Administrador)	49

LISTA DE ANEXOS

ANEXO A ENCUESTA REALIZADA PARA LOS REQUERIMIENTOS DE LA APLICACIÓN.....	70
ANEXO B CUESTIONARIO DE USABILIDAD DE LA APLICACIÓN AGENDA MÉDICA.....	74

RESUMEN

Título:

AGENDA MÉDICA PERSONAL PARA DISPOSITIVOS MÓVILES.¹

Autores: CAICEDO ORTIZ, Geraldine, LEÓN MÉNDEZ, Laura Milena.²

Palabras Claves:

Nube, Desarrollo Móvil, Prototipos, Android, Agenda, Medicamento, Fórmula.

DESCRIPCIÓN:

Ya que hoy en día se tienen tantas tareas por realizar, compromisos que cumplir; se tiende a olvidar algo tan básico pero importante como lo es asistir a una cita médica, a un control, un examen. Incluso es fácil olvidar la hora de la ingesta de los medicamentos recetados o se acostumbra a tomar a la hora en que se recuerda, lo cual no es correcto ya que cada medicamento tiene su horario para que pueda hacer efecto y así ayudar de manera positiva a la hora de tratar enfermedades y poder llegar a una cura o a un óptimo tratamiento que mejore el estilo de vida.

Por esta razón se desarrolló una agenda médica personal para dispositivos móviles que ayuda a la hora de recordar: citas médicas: el lugar, la hora, la fecha de dicha cita. En la toma de exámenes aparte de hora y fecha también recordar que requisitos son necesarios para poder realizarlos y en lo referente en ingesta de medicamentos avisar la hora exacta que estos deben ser tomados o aplicados; además de presentar una serie de información de los medicamentos que sean necesitados, y llevar un registro de aquellos profesionales médicos que han atendido en algún momento.

Esta agenda fue desarrollada siguiendo las metodologías Kanban, programación extrema, y utilizando el framework PhoneGap junto a los lenguajes HTML, JavaScript, JQueryMobile y Ajax; todo esto para el sistema operativo Android, también se pobló una base de datos con la información pertinente respecto a los medicamentos que generalmente son utilizados en los diferentes tratamientos de los usuarios.

¹ Trabajo de grado con modalidad Trabajo de Investigación

²Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniera de Sistemas e Informática Director Mg. Pedro Javier Trujillo Tarazona.

ABSTRACT

Title:

PERSONAL MEDICAL AGENDA FOR MOBILE DEVICES ³

Authors: CAICEDO ORTIZ, Geraldine, LEÓN MÉNDEZ, Laura Milena⁴

Keywords:

DESCRIPTION:

In today's world there are so many tasks to be done and commitments to fulfill; People tend to forget basic yet important things, such as doctor's appointment or even the correct time to take their medication pills. Taking your pills at the right time of day can make a huge difference in how well they work. Because if people take pills in a correct way, it can help to treat diseases, to find a cure or to lead a good lifestyle.

For this reason, a mobile application has been developed that helps you remember the place, the time and the date of your next doctor's appointment. Also with the exam stuffs and medication things. The application will remind you of any necessary requirements needed for your next doctor's appointment In terms of medication intake. The application will inform you of the exact time your pills should be taken and provide further information on the medication. The application will also keep track of your visits to doctors and pharmacies.

This agenda was developed following Kanban methodologies, eXtreme Programming, and using the PhoneGap framework along with the languages HTML, JavaScript, JQueryMobile and Ajax; all this for the Android operating system. Also, a database was populated with relevant information regarding the medicines that are generally used in the different treatments of users.

³Degree Work with modality Investigation Work

⁴ Faculty of Physical-Mechanical Engineering, Systems Engineering School, Director MSc. Pedro Javier Trujillo Tarazona

GLOSARIO

AGENDA: Libro, cuaderno o dispositivo electrónico en que se apunta, para no olvidarlo, aquello que se ha de hacer ya sean reuniones, citas o diferentes actividades por realizar.

APLICACIÓN HÍBRIDA: Generalmente consisten en Apps que contiene en su interior el navegador web del dispositivo. Para su desarrollo se utilizan frameworks de desarrollo basados en lenguajes de programación web (HTML, CSS y JS).

APLICACIÓN NATIVA: Aplicación móvil que se desarrolla directamente en el lenguaje nativo de cada terminal. Por eso, si se va a desarrollar una aplicación nativa se tiene que utilizar un lenguaje diferente para cada Sistema Operativo:

iOS: Objective C

Android: Java

Windows: C# y Visual Basic .NET.

BlackBerry 10: C++

ANDROID: Sistema operativo que fue creado especialmente para teléfonos con pantalla táctil, los llamados de nueva generación o los inteligentes, las tablets comunes y las que funcionan con líneas telefónica; entrando en esta gama los relojes inteligentes, televisores y algunos aditamentos de los nuevos automóviles.

BROWSER: Es un software, aplicación o programa que permite el acceso a la Web, interpretando la información de distintos tipos de archivos y sitios web para que estos puedan ser visualizados.

FRAMEWORK:(En español infraestructura) es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software.

HARDWARE: Partes físicas tangibles de un sistema informático; sus componentes eléctricos, electrónicos, electromecánicos y mecánicos. Cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado componen el hardware.

PROGRAMACIÓN ORIENTADA A OBJETOS: Es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos manipulan los datos de entrada para la obtención de datos de salida específicos, donde cada objeto ofrece una funcionalidad especial. Muchos de los objetos pre-diseñados de los lenguajes de programación actuales permiten la agrupación en bibliotecas o librerías, sin embargo, muchos de estos lenguajes permiten al usuario la creación de sus propias bibliotecas.

MÉTODO CLÍNICO: Proceso o secuencia ordenada de acciones que los médicos han desarrollado para generar su conocimiento desde el comienzo de la era científica. Es el método científico aplicado a la práctica clínica; es el orden recorrido para estudiar y comprender el proceso de salud y de enfermedad de un sujeto en toda su integridad social, biológica y psicológica.

MOCKUPS: Maqueta es un modelo a escala o tamaño real de un diseño o un dispositivo, utilizado para la demostración, evaluación del diseño, promoción, y para otros fines.

MULTIHILO: Un programa multihilo contiene dos o más partes que se pueden ejecutar de manera concurrente o simultánea. A cada una de esas partes se les denomina hilo y cada uno de ellos define un camino de ejecución independiente. Se puede ver a la programación multihilo como una forma especializada de multitarea.

MULTIUSUARIO: En general se le llama multiusuario a la característica de un sistema operativo o programa que permite proveer servicio y procesamiento a múltiples usuarios simultáneamente, estrictamente es pseudo-simultáneo (tanto en paralelismo real como simulado).

MULTIPLATAFORMA: Tributo conferido a programas informáticos o métodos y conceptos de cómputo que son implementados e interoperan en múltiples plataformas informáticas.

COMUNICACIÓN ASINCRÓNICA: Se denomina comunicación asincrónica al proceso comunicativo que se lleva a cabo sin coincidencia temporal. Esto quiere decir que la emisión y la recepción de los mensajes están separadas por un cierto periodo de tiempo.

WEB VIEW: Aplicación proporcionada por Android a la que los desarrolladores tienen acceso a la hora de programar sus aplicaciones.

ACRÓNIMOS

API: *Application Programming Interface* (interfaz de programación de aplicaciones).

AJAX: *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML) aunque hoy día ya no es una tecnología ligada a XML con lo cual no pueden asociarse las siglas a estos términos.

CGI: Common Gateway Interface (Interfaz de entrada común).

CLI: Command-line Interface (interfaz de línea de comandos o interfaz de línea de órdenes).

CRES: Comisión de Regulación de Salud.

CSS: Cascading StyleSheets (Hojas de estilo en cascada).

CTC: Comité Técnico Científico.

EPS: Empresas Promotoras de Salud.

HTML: HyperText Markup Language (lenguaje de marcas de hipertexto).

IPS: Institución Prestadora de Salud.

OMS: Organización Mundial de la Salud.

PHP: Hypertext Preprocessor (pre-procesador de hipertexto).

POS: Plan Obligatorio de Salud.

MVC: Modelo Vista Controlador.

NO POS: Aquellos servicios que no entran en el Plan Obligatorio de Salud POS.

SGSSS: Sistema General de Seguridad Social en Salud.

SDK: Software Development Kit (kit de desarrollo de software).

TIC: Tecnologías de la Información y la Comunicación.

XP: eXtreme Programming (Metodología programación extrema).

INTRODUCCIÓN

Desde tiempos inmemorables, el deseo de las personas por estar sanas y llevar una vida plena y sin complicaciones va aumentando de la mano con la tecnología la cual va avanzando día a día intentando eliminar cada uno de los problemas por completo y así todas las amenazas que estos representan para la vida cotidiana, dando solución a estos problemas, los cuales la mayoría de veces tienen un alto nivel de complejidad, ayudándose de entre otras cosas de lo que en estos momentos ofrece la alta capacidad computacional, convirtiéndose así en objetivo de científicos e ingenieros. Esta meta propuesta ha llevado al desarrollo de componentes hardware y software que han hecho posible grandes avances que van desde implantes hechos de materiales totalmente diferentes a los órganos comunes hasta aplicaciones que le facilitan las cosas tanto a la unidad médica en general como al paciente.

Tomando los recursos informáticos que brinda el alma mater y teniendo presente las necesidades que se presentan en las vidas de todas las personas del entorno en este trabajo, se pretende presentar una aplicación que le facilitará la vida a todas aquellas personas que no solo tienen que estar pendiente de su bienestar sino el de su entorno familiar, teniendo en cuenta que todos son pacientes y que cada vez que se presenta algún inconveniente que afecta directamente la salud son una cantidad de cosas de las que se tiene que estar pendiente para llegar a un óptimo estado de salubridad.

El modelo aquí presentado, es una aplicación que le facilitará la vida a todas las personas no solo de esta universidad sino en general, tiene una arquitectura diseñada en base a varias encuestas realizadas a más de cien prospectos, cabe resaltar que no solo se quedara en esta versión sino se espera ir mejorando el producto conforme a las necesidades que el cliente va informando y a su vez no solo se quedara en un solo sistema operativo sino se espera llegar a ser una aplicación multiplataforma.

A continuación se presentara una recopilación del trabajo realizado para generar el producto final, el software que fue probado, dando información sobre los lenguajes de programación usados, así como los motores de bases de datos con los cuales se gestionó toda la información pertinente de los usuarios y la metodología que hizo esta aplicación posible.

Dando a conocer la importancia de tener bien definidos los casos de uso del software así como un correcto modelo de funcionamiento para la guía hacia el objetivo principal. Se podrá observar los procesos que fueron llevados a cabo no solo en el desarrollo del producto, sino también en todo lo que esto implicó, dando como resultado final un software que facilitara la vida a muchos usuarios.

1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

Al considerar el ámbito local de las necesidades de las personas en cuanto a la cantidad de información que debe tener pendiente para su desempeño social, lo cual se dificulta más con el pasar de los años, y que se presenta crítico en lo concerniente a la retención de información detallada e incluso básica para el cuidado de su salud, y que además se complica cuando una persona debe prestarle atención a los compromisos médicos o pertinentes a la salud de varias personas en su entorno familiar; y al tomar en cuenta que actualmente se dispone de tecnologías generalmente subutilizadas, se pretende crear una aplicación móvil multifuncional que permita tener a un toque de distancia una práctica herramienta compuesta básicamente por una agenda personal médica que además de facilitar información sobre una cita también les recuerda los compromisos de ingesta oportuna y bien dosificada de medicamentos.

Este proyecto propone la elaboración de software que podrá ser utilizado por cualquier persona que esté en condiciones de manipular un teléfono celular inteligente o medianamente inteligente, para realizar las tareas y actividades antes señaladas en el control en tiempo real de la agenda del cuidado de la salud de las personas a cargo o responsabilidad del usuario de dicho teléfono.

El ámbito tecnológico de la telefonía celular y la computación en la nube tiene un gran avance y un sinnúmero de oportunidades para desarrollar aplicaciones, una de ellas es la que se aborda. Hay otras aplicaciones que sirven para recordar algunas de las cosas que son necesarias en la toma de medicamentos y analizan los componentes de estos mismos, así como hay aplicaciones para medir el desempeño de diferentes componentes del organismo; por igual, otras aplicaciones sirven para llevar una dieta y valorar los alimentos y su aporte en niveles de calorías y lo como estos afectaran al cuerpo; todas estas aplicaciones son pagas y esta propuesta se orienta a ofrecerle a la comunidad una herramienta que les ayude en la gestión de las actividades y el tiempo dedicado al cuidado de su salud en los aspectos esenciales.

En conclusión, al observar diversos avances que día a día se están logrando con las tecnologías de la información y la comunicación TIC se debe aprovechar estos recursos para mejorar las condiciones de vida, otorgándoles a los usuarios mecanismos que le ayuden en el cuidado de su salud, no tener que estar pendiente de un reloj y un calendario para saber la fecha y hora de una cita o para

la ingesta de medicamentos, con la facilidad de saber con claridad y rápidamente cual medicina es la que tiene que tomar, que es lo que tiene que llevar a la cita médica, y otros detalles que han de priorizar para agilizar la gestión de la información del cuidado de la salud asociada a la agenda personal de carácter médico.

De otra parte, este desarrollo también podrá ayudar, de forma indirecta, a entidades como las Entidades Promotoras de Salud EPS y las Instituciones Prestadoras de Servicios IPS, a reducir las tasas de incumplimiento y pérdida de citas. Lo cual implicaría un aprovechamiento más eficiente de los recursos de la salud. El mejorar las condiciones de vida, otorgándoles a los usuarios mecanismos que le ayuden en el cuidado de su salud, ya que se observa que siempre se está pendiente de un reloj y un calendario para saber la fecha y hora de una cita o para la ingesta de medicamentos, se busca darle la facilidad de saber con claridad y rápidamente cual medicina es la que tiene que tomar, que es lo que tiene que llevar a la cita médica, y otros detalles que han de priorizar para agilizar la gestión de la información del cuidado de la salud asociada a la agenda personal de carácter médico.

2. OBJETIVOS

2.1. OBJETIVO GENERAL

Desarrollar una agenda personal médica para dispositivos móviles que apoye al usuario en el control de citas médicas, eventos de tratamientos y administración de medicamentos.

2.2. OBJETIVOS ESPECÍFICOS

- Identificar necesidades esenciales de usuarios para la agendación de citas y control de ingesta de medicamentos.
- Caracterizar las necesidades de los usuarios respecto de una herramienta para agendación de citas y control de ingesta de medicamentos.
- Diseñar la agenda personal médica para dispositivos móviles siguiendo los requerimientos de los usuarios.
- Desarrollar la aplicación móvil soportada en un servicio implementado en la nube.
- Validar y ajustar la herramienta en diferentes escenarios de prueba.

3. MARCO TEÓRICO

3.1. SALUD

La salud es un estado de completo bienestar físico, mental y social, no solamente la ausencia de enfermedad o dolencia, según la definición presentada por la Organización Mundial de la Salud (OMS) en su constitución aprobada en 1948. Este concepto se amplía a: "La salud es un estado de completo bienestar físico, mental y social, y no solamente la ausencia de afecciones o enfermedades". En la salud, como en la enfermedad, existen diversos grados de afectación y no debería ser tratada como una variable dicotómica. Así, se reformularía de la siguiente manera: "La salud es un estado de bienestar físico, mental y social, con capacidad de funcionamiento, y no sólo la ausencia de afecciones o enfermedades". También puede definirse como el nivel de eficacia funcional o metabólica de un organismo tanto a nivel micro (celular) como a nivel macro (social). En 1992 un investigador amplió la definición de la OMS, al agregar: "y en armonía con el medio ambiente"⁵

3.1.1. Medicamentos

Un medicamento es uno o más fármacos, integrados en una forma farmacéutica, presentado para expendio y uso industrial o clínico, y destinado para su utilización en las personas o en los animales, dotado de propiedades que permitan el mejor efecto farmacológico de sus componentes con el fin de prevenir, aliviar o mejorar el estado de salud de las personas enfermas, o para modificar estados fisiológicos.

Un medicamento es una sustancia o preparado que tiene propiedades curativas o preventivas, se administra a las personas o a los animales y ayuda al organismo a recuperarse de los desequilibrios producidos por las enfermedades o a protegerse de ellos.⁶

⁵ Organización Mundial de la Salud. (s.f). Salud mental (Definición). Recuperado de http://www.who.int/topics/mental_health/es/

⁶ Colegio de Farmacéuticos de la Provincia de Barcelona. (s.f). Subdepartamento de Registro (Definición de Medicamentos). Recuperado de http://www.ispch.cl/anamed/subdeptoregistro/definicion_medicamentos

3.1.2. Plan Obligatorio de Salud (POS)

Son aquellos servicios básicos al cual los usuarios tiene derecho a lo referente en atención de salud en el Sistema General de Seguridad Social de Colombia (SGSSS), cuya función es la protección de las salud, la prevención, curación de enfermedades y el suministro de medicamentos para el afiliado y su grupo familiar estos servicios también se complementan con el reconocimiento de las prestaciones económicas en los casos de incapacidad por enfermedad o en caso de licencia por maternidad.⁷

Los servicios entregados en el POS se encuentran regulados por el Estado mediante el Acuerdo 029 del 2011 de la comisión de regulación de salud (CRES)⁸.

3.1.3. Medicamentos POS

Son aquellos medicamentos a los cuales tienen derecho todos los afiliados al Sistema General de Seguridad Social en Salud en Colombia.

Según el listado de medicamentos y servicios que aprobó el Ministerio de Salud.⁹

3.1.4. Medicamentos NO POS

Son aquellos medicamentos que no están incluidos en el Manual de medicamentos (Acuerdos 228 y 263 de 2002, 236 y 282 de 2004 y 336 de 2006), por lo cual el usuario no tiene acceso a este por el Sistema General de Seguridad Social en Salud en Colombia; si el médico especialista formula un medicamento NO POS, para tratar un diagnóstico que esté cubierto o que no esté cubierto en el POS, se requiere autorización del Comité Técnico Científico CTC de la entidad. En este caso la IPS envía a la entidad: formato de solicitud procedimientos y medicamentos NO POS diligenciado por el médico tratante, y datos contenidos en documento de identidad.

⁷ COLOMBIA. CONGRESO DE LA REPÚBLICA. Ley 100. (23, diciembre, 1993). Por la cual se crea el sistema de seguridad social integral y se dictan otras disposiciones. Diario Oficial. Bogotá, D.C., 1993. no. 41148. p. 1-168.

⁸ COMISION DE REGULACIÓN DE SALUD. Acuerdo número 29. (28 de Diciembre del 2011). Por el cual se sustituye el Acuerdo 028 de 2011 que define, aclara y actualiza integralmente el Plan Obligatorio de Salud. Recuperado de <https://www.minsalud.gov.co/salud/pos/mi-plan/Normatividad%20y%20estudios%20de%20mi%20plan/Acuerdo%20029%20pos%202012.pdf>

⁹ MINISTERIO DE SALUD. Listados actualizados de medicamentos, procedimientos, y servicios incluidos tanto en el POS del régimen contributivo como en el POS del régimen subsidiado(Sitio web del Ministerio de Salud y Protección Social. <https://www.minsalud.gov.co/salud/Paginas/pos.aspx>)

3.1.5. Consultas Médicas

Es el tiempo que el médico dedica al paciente donde le realizara exámenes físico e interrogara con respecto a sus antecedentes clínicos y los síntomas que presenta, así el médico puede formular una hipótesis que luego buscara confirmar ya sea con diferentes tipos de exámenes para así llegar a una conclusión y buscar el tratamiento adecuado; durante todo este proceso el médico le programara diferentes citas médicas para ver el desarrollo del paciente.

3.1.6. Exámenes Médicos

Son parte fundamental del método clínico (diferentes etapas que el médico aplica al paciente para poder llegar a un diagnóstico y así seguir un tratamiento), es donde el médico realiza pruebas al paciente cuyo resultado reflejara las afecciones que pueda estar sufriendo y así poder formular un tratamiento adecuado y eficaz que el paciente seguirá para eliminar o disminuir estas afecciones.

3.2. TECNOLOGÍAS INVOLUCRADAS EN EL DESARROLLO DEL PROYECTO

3.2.1. JavaScript

Es uno de los lenguajes que da muchas posibilidades ya que con él se puede hacer todo tipo de acción e interacción; todos los navegadores web modernos y la mayoría de sitios web incluyen intérpretes de JavaScript. Este lenguaje se adapta bien a los estilos de programación orientada a objetos también se caracteriza por ser basado en acciones con menos restricciones.

“JavaScript es parte de la Tríada de tecnologías que todos los desarrolladores web deben aprender: HTML para especificar el contenido de páginas web, CSS para especificar la presentación de páginas web y JavaScript para especificar el comportamiento de las páginas web”¹⁰

¹⁰ Flanagan, David. *JavaScript; The Definitive Guide* (Gravenstein Highway North, O'Reilly Media, 2011). Pag:1-3

3.2.2. JQuery

Es una biblioteca gratuita de JavaScript de fácil uso y gran potencia; ayuda a simplificar las tareas a la hora de la creación de páginas web, otra de sus grandes ventajas es que gracias a esta biblioteca cuando se escribe código JavaScript este va a ser capaz de ejecutarse casi sin errores en cualquier navegador sin importar su antigüedad. También debido a que JQuery permite programar diferentes funcionalidades por medio de plugins (complementos) entonces se puede lograr la validación de formularios, interfaces de usuario bien desarrolladas, plantillas etc. En conclusión, jQuery agrega una gran cantidad de funciones nuevas a JavaScript, los cuales podrán ser utilizados en diferentes sitios web.

3.2.3. Hypertext Pre-Processor (PHP)

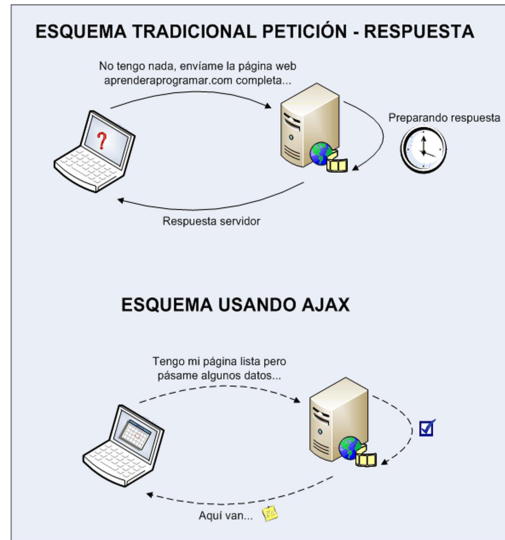
Es un lenguaje de lado del servidor caracterizado por su potencia y simplicidad también tiene un soporte generalizado en la mayoría de los servidores hosting incluso en los más simples y económicos. Este lenguaje se caracteriza por su facilidad porque permite incorporar fragmentos de códigos mínimos dentro de una página común con HTML; PHP ofrece una gran cantidad de funciones pero una de las más destaca son las bases de datos ya que permite establecer una conexión con ella, por lo cual un ejemplo es que permite mostrar en la pantalla los datos del cliente cuando sea requerida.¹¹

3.2.4. Asynchronous JavaScript y XML (AJAX)

Es una técnica de programación que combina varios lenguajes de programación para crear aplicaciones interactivas. Para entender cómo funciona AJAX se tiene la siguiente gráfica:

¹¹ Alvarez, Miguel Angel. (s.f). Desarrollo web (Manual de PHP). Recuperado de <http://www.desarrolloweb.com/manuales/manual-php.html#capitulos12>

Figura 1 Diferencia de petición-respuesta del modo tradicional vs al usar AJAX¹²



Rodríguez Rancel, Mario. (s.f). Aprender a programar (Imagen). Recuperado de http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=882:i-que-es-y-para-que-sirve-ajax-ventajas-e-inconvenientes-javascript-asincrono-xml-y-json-cu01193e&catid=78:tutorial-basico-programador-web-javascript-desde-&Itemid=206

Ya que AJAX es una tecnología asincrónica lo que quiere decir que los datos adicionales que se solicitan en el servidor se cargan en segundo plano así no interfieren en la visualización ni en el comportamiento de la página. Por lo tanto la figura anterior muestra que a la hora de hacer una petición de una página web al servidor de manera tradicional este se toma su tiempo y envía la respuesta de forma completa en cambio a la hora de usar AJAX se tiene una página web cargada le dice al servidor que envíe algunos datos faltantes, el servidor procesa la petición enviando la respuesta de forma más rápida ya que no tiene que elaborar de nuevo la página web completa solo los datos faltantes.

3.2.5. Lenguaje de Marcado de Hipertexto (HTML)

Es el elemento más básico a la hora de crear una página web. El estándar HTML lo define la W3C (World Wide Web Consortium) y actualmente HTML se encuentra en su versión HTML5.

¹² Rodríguez Rancel, Mario. (s.f). Aprender a programar (Imagen). Recuperado de http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=882:i-que-es-y-para-que-sirve-ajax-ventajas-e-inconvenientes-javascript-asincrono-xml-y-json-cu01193e&catid=78:tutorial-basico-programador-web-javascript-desde-&Itemid=206

Ya que HTML no tiene funciones aritméticas ni variables de control propias como en los lenguajes de programación, no pertenece a este grupo pero eso no quiere decir que no se pueda usar en conjunto con cualquier otro lenguaje de programación para crear páginas web ya que este ayuda a describir la estructura básica de la página y así organizar la forma en cómo se mostrara su contenido.

HTML es un lenguaje de marcado descriptivo que se escribe en forma de etiquetas para definir la estructura de una página web y su contenido como texto, imágenes, entre otros, de modo que HTML es el encargado de describir (hasta cierto punto) la apariencia que tendrá la página web.¹³

3.2.6. Hojas de Estilo en Cascada (Cascading Style Sheets (CSS))

Este lenguaje nos ayuda a separar lo que es la estructura del documento HTML de su presentación; las hojas de estilo tienen muchas propiedades que nos ayudan a diseñar la presentación estética que tendrá una página web; esto da la ventaja de poder controlar de manera eficiente el documento y sus atributos por lo tanto el HTML será un documento más versátil y liviano.

Cuando se utiliza CSS, la etiqueta -H1- no debería proporcionar información sobre cómo va a ser visualizado, solamente marca la estructura del documento. La información de estilo previamente separada en una hoja de estilo CSS, especifica cómo se ha de mostrar -H1- : color, fuente, alineación del texto, tamaño, entre otras características.

3.2.7. MySQL

Es un sistema de administración de bases de datos multihilo y multiusuario esto quiere decir que puede ser usado por varios usuarios al mismo tiempo.

Gracias a que este gestor permite realizar cambios en sus sitios de manera simple sin necesidad de cambiar todo el código web sino solo cambiar un archivo, es muy usado en desarrollo web.

MySQL trabaja como un sistema centralizado de gestión de datos.

A la hora de combinarse con PHP es utilizada para realizar aplicaciones cliente/servidor ya que es necesaria una base de datos segura, rápida y potente.¹⁴

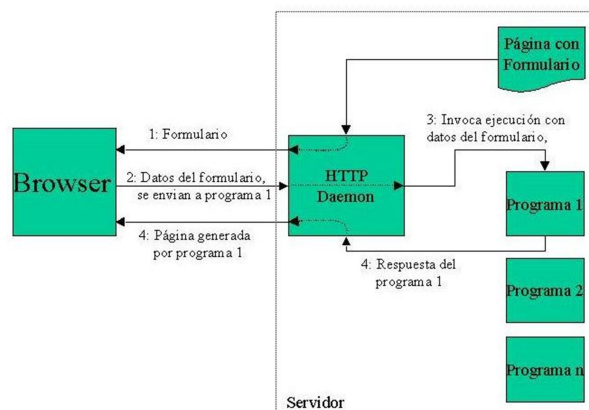
¹³ Romero García, Israel. (2013). Acerca de HTML (Manual de HTML). Recuperado de <http://www.acercadehtml.com/manual-html.html>

¹⁴ Fritzler ,Danilo. (2008). Culturación (Que es y Para qué Sirve MySQL). Recuperado de <http://culturacion.com/que-es-y-para-que-sirve-mysql/>

3.2.8. Common Gateway Interface (CGI)

Es un método que ayuda a transmitir información a un compilador que está instalado en un servidor; es muy utilizado a la hora de utilizar contadores, bases de datos, motores de búsqueda, generadores de email automático, etc.

Figura 2 Esquema de Ejecución CGI¹⁵



Universidad de Concepción. (s.f). Common Gateway Interface CGI. Recuperado de <http://www.inf.udec.cl/~jlopez/DSWR/cgi.html>

La figura muestra que cada vez que se envían los datos de un formulario se crea un proceso en el servidor para ejecutar el programa en cuestión, por lo cual genera una respuesta que es enviada a través del servidor Http al navegador (navegador de internet), luego se elimina el proceso creado. Los programas CGI acceden a las bases de datos, para acceder a los datos desde su localización y ser presentados para luego ser modificados a través de páginas web.

3.2.9. PhoneGap

Es un framework utilizado en el desarrollo de aplicaciones móviles; ya que cada plataforma móvil ya sea: Android, iPhone o Windows; requieren diferentes lenguajes de programación a la hora de crear aplicaciones móviles, PhoneGap soluciono esto ya que permite a los programadores de dispositivos móviles utilizar herramientas como JavaScript, HTML5 y CSS3. Por lo cual las aplicaciones resultantes son híbridas lo que quiere decir que no son nativas a la plataforma.

¹⁵ Universidad de Concepción. (s.f). Common Gateway Interface CGI. Recuperado de <http://www.inf.udec.cl/~jlopez/DSWR/cgi.html>

PhoneGap maneja API (Application Programming Interface) que permite tener acceso a elementos como el acelerómetro, la cámara, los contactos en el dispositivo, la red, el almacenamiento, las notificaciones, etc. Estas API se conectan al sistema operativo usando el código nativo del sistema huésped a través de una Interfaz de funciones foráneas en JavaScript.

3.2.10. Apache Cordova

Es un marco de desarrollo móvil de código abierto. Este marco permite utilizar las tecnologías estándar web como HTML5, CSS3 y JavaScript para desarrollo multiplataforma por lo tanto este tipo de aplicación móvil se le denomina aplicación híbrida, esto ayuda a evitar el lenguaje de desarrollo nativo de cada plataforma móvil. Las aplicaciones se ejecutan dentro de envolturas para cada plataforma y dependen de enlaces estándares API para acceder a cada dispositivo ya sean sensores, datos y estado de la red.

Usar Apache Cordova si eres:

- Desarrollador que desea extender una aplicación a través de más de una plataforma, sin tener que re implementarlo con herramienta y lenguaje de cada plataforma.
- Desarrollador web y desea implementar una aplicación web que se envasa para su distribución en varias portales *appstore*.
- Desarrollador móvil interesado en que se mezclan los componentes de la aplicación nativa con un *WebView* (ventana del navegador especial) que puede tener acceso a las API del dispositivo, o si quiere desarrollar una interfaz plugin entre componentes *WebView* y nativos.¹⁶

3.2.10.1. Componentes Básicos Apache Cordova

Las aplicaciones en Apache Cordova se basan en un archivo común *config.xml*, que proporciona información acerca de la aplicación y especifica los parámetros que afectan el cómo va a funcionar, como si responde a la orientación entonces cambia de puesto. Este archivo se adhiere a la especificación de Empaquetado de la aplicación Web, *widget*, o de la W3C.

La misma aplicación se implementa como una página web, un archivo local llamado *index.html*, que hace referencia a cualquier CSS, JavaScript, imágenes,

¹⁶ Cordova (Documentación Apache Cordova). (s.f.). Recuperado de <https://cordova.apache.org/docs/es/latest/guide/overview/>

archivos multimedia u otros recursos que son necesarios para que se ejecute de forma predeterminada. La aplicación se ejecuta como un *WebView* dentro de la envoltura de la aplicación nativa.

El *WebView* Cordova habilitado puede proporcionar la aplicación con su interfaz de usuario completa. En algunas plataformas, también puede ser un componente dentro de una aplicación híbrida más grande, que mezcla la vista Web con componentes de la aplicación nativa. Una interfaz *plugin* está disponible para Cordova y componentes nativos para comunicarse con los demás. Esto permite invocar un código de JavaScript. Idealmente, las API de JavaScript para ese código nativo son consistentes a través de múltiples plataformas de dispositivos.

Cordova proporciona el tiempo de ejecución en el cual los pueden ejecutar. Si se desea utilizar widgets UI o un marco MV *, se necesitará seleccionar aquellos e incluirlos en la solicitud como material de terceros.

3.2.10.2. Vías de desarrollo.

Se puede utilizar dos flujos de trabajo básicos para crear una aplicación móvil. Mientras que a menudo se puede utilizar cualquier flujo de trabajo para realizar la misma tarea, cada uno de ellos ofrece ventajas:

- **Flujo de trabajo multiplataforma (CLI):** Este flujo de trabajo si la aplicación es para ejecutarse en los diferentes sistemas operativos móviles, con poca necesidad específica de la plataforma desarrollo. Este flujo de trabajo se centra en la utilidad Cordova, también conocido como *CLI*. El CLI es una herramienta de alto nivel que permite construir proyectos para muchas plataformas a la vez, muy lejos de la funcionalidad de scripts de shell de bajo nivel de abstracción. La CLI copia un conjunto común de web activos en subdirectorios para cada plataforma móvil, hace que cualquier cambio de configuración se realice en cada uno. La CLI también proporciona una interfaz común para aplicar plugins para la aplicación.
- **Flujo de trabajo centrado en plataforma:** Este flujo de trabajo se usa cuando se desea concentrar en construir una aplicación para una sola plataforma y hay necesidad de poder modificarlo en un nivel inferior. Se tiene que utilizar este enfoque cuando se quiere que la aplicación mezcle los componentes nativos personalizados con componentes Cordova basados en web. Como regla general, se utiliza este flujo de trabajo si se necesita modificar el proyecto dentro del SDK. Este flujo de trabajo se basa en un conjunto de scripts de shell de nivel inferior que se adaptan para cada plataforma soportada y una utilidad de Plugman separada que le permite aplicar plugins. Mientras este flujo de

trabajo puede utilizar para crear aplicaciones multiplataforma, es generalmente más difícil porque la falta de una herramienta de alto nivel significa construir diferentes ciclos y modificaciones de plugin para cada plataforma.

3.2.10.3. Pluggins de Cordova necesarios para la aplicación

BatteryStatus: Este evento se desencadena cuando cambia el porcentaje de carga de la batería en menos de 1 por ciento, o si el aparato está enchufado o desenchufado.

El controlador del estado de batería se pasa a un objeto que contiene dos propiedades:

- **nivel:** el porcentaje de carga de la batería (0-100). (*Número*)
- **isPlugged:** un valor booleano que indica si el dispositivo está conectado plug (*Boolean*)

Las aplicaciones normalmente deben utilizar *window.addEventListener* para conectar un detector de eventos una vez que se desencadene el evento [*deviceready*] (*events.deviceready.html*).

Device

El objeto *device* describe del dispositivo el hardware y software.

Propiedades

device.model

device.cordova

device.platform

device.uuid

device.version

device.name

Ámbito de variable

Puesto que el *device* se asigna al objeto de *window*, es implícitamente en el ámbito global.

DeviceMotion

Captura de movimiento del dispositivo en la dirección x, y y z.

Métodos

accelerometer.getCurrentAcceleration

accelerometer.watchAcceleration

accelerometer.clearWatch

Argumentos

AccelerometerSuccess

AccelerometerError

AccelerometerOptions

Objetos (sólo lectura)

Acceleration

DeviceOrientation (Brújula)

Obtiene la dirección que apunta el dispositivo.

Métodos

compass.getCurrentHeading

compass.watchHeading

compass.clearWatch

Argumentos

compassSuccess

compassError

compassOptions

compassHeading

Acceso a la función

Cordova implementa el nivel de dispositivo APIs como *plugins*. El uso del comando CLI *plugin*, que se describe en la interfaz de línea de comandos, para añadir o eliminar esta característica para un proyecto:

```
$ cordova plugin add org.apache.cordova.device-orientation
```

```
  $ cordova plugin ls
```

```
['org.apache.cordova.device-orientation']
```

```
$ cordova plugin rm org.apache.cordova.device-orientation
```

Notificación (Dialogs)

Notificaciones de dispositivo audible, visual y táctil.

Acceso a la función

Línea de comandos, para añadir o eliminar esta característica para un proyecto:

```
$ cordovapluginadd org.apache.cordova.dialogs
```

```
$ cordovapluginadd org.apache.cordova.vibration
```

```
  $ cordovapluginls
```

```
['org.apache.cordova.dialogs',
```

```
'org.apache.cordova.vibration']
```

```
  $ cordovapluginrm org.apache.cordova.dialogs
```

```
$ cordova plugin rm org.apache.cordova.vibration
```

File

Archivo

Una API para leer, escribir y navegar por las jerarquías de sistema de archivo, basadas en la api de archivo w3c.

Objetos

DirectoryEntry

DirectoryReader

Archivo

FileEntry

FileError

FileReader

FileSystem

FileTransfer

FileTransferError

FileUploadOptions

FileUploadResult

FileWriter

Flags

LocalFileSystem

Metadata

Acceso a la función

Línea de comandos, para añadir o eliminar esta característica para un proyecto:

```
$ cordova plugin add org.apache.cordova.file
```

```
$ cordova plugin ls
```

```
  [ 'org.apache.cordova.file' ]
```

```
$ cordova plugin rm org.apache.cordova.file
```

Conexión

El objeto de *connection*, expuesto a través de *navigator.connection*, proporciona información sobre conexión celular y wifi del dispositivo.

Propiedades

connection.type

Constantes

Connection.UNKNOWN

Connection.ETHERNET

Connection.WIFI

Connection.CELL_2G

Connection.CELL_4G

Connection.CELL_3G

Connection.CELL

Connection.NONE

Acceso a la función

Línea de comandos, para añadir o eliminar esta característica para un proyecto:

```
$ cordova plugin add org.apache.cordova.network-information
```

```
  $ cordova plugin ls
```

```
[ 'org.apache.cordova.network-information' ]
```

```
  $ cordova plugin rm org.apache.cordova.network-information17
```

¹⁷ Cordova (Documentación Apache Cordova Para Android). (s.f.). Recuperado de <https://cordova.apache.org/docs/en/latest/guide/platforms/android/index.html>

3.2.11. Modelo Vista Controlador (MVC)

El modelo vista controlador MVC, es un patrón de arquitectura software se compone de tres componentes que son el modelo, la vista y el controlador, esto para darle al usuario final una aplicación agradable en interacción y secuencias.

Modelo: contiene toda la información almacenada en la base de datos junto con la reglas de negocio encargada de alterar la información según acciones del usuario.

Vista: es la presentación del modelo en una forma ordenada y presentable para el usuario.

Controlador: código que obtiene datos dinámicamente y son generados para la vista al ser requeridos por el usuario.

El MVC funciona así: el usuario realiza una acción, el controlador interpreta la entrada del usuario, generando un mensaje de acción al modelo que le retorna una respuesta, seguido de esto el controlador envía a la vista, la vista toma los datos y los deja listos para entregar de nuevo.

De esta forma se interactúa a través del HTML, JavaScript, el usuario, con el php y las peticiones Ajax van interactuando con la base de datos.

4. METODOLOGÍA

Se propuso como metodología de desarrollo de software ágil de este proyecto la Programación extrema (XP)¹⁸ y usando algunos de los principios de la metodología Kanban¹⁹.

Se decidió aplicar una integración de estas metodologías ágiles debido a los aportes que le dieron al proyecto. En primer lugar se presentará lo que fue tomado de Kanban; como metodología ágil de gestión de proyectos tiene las ventajas de que es muy fácil de usar, fácil de actualizar, fácilmente asimilable por el equipo, muy visual, permite ver el estado de todos los proyectos en un golpe de vista y ayuda a decidir en qué trabajar cada día/semana.

En cuanto a la programación extrema la cual es basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas y que da prioridad a los trabajos que dan un resultado directo se ha decidido implementar todo el contexto de su metodología y así ver paso a paso como va a ser la evolución del proyecto. El objetivo que se perseguía en el momento de crear esta metodología era la búsqueda de un método que hiciera que los desarrollos fueran más sencillos, aplicando el sentido común.

¹⁸**XP (programación extrema):** metodología ágil que tiene como principal objetivo aumentar la productividad a la hora de desarrollar un proyecto software. Da prioridad a los trabajos que dan un resultado directo. Leer más en (Pérez Esteso, Mario. (s.f). GeekyTheory (Programación extrema). Recuperado de <https://geekytheory.com/programacion-extrema-que-es-y-principios-basicos>).

¹⁹ **Kanban:** es una metodología ágil para gestionar el trabajo intelectual, con énfasis en la entrega justo a tiempo, mientras no se sobrecarguen los miembros del equipo. Leer más en (Álvarez, Miguel Angel. (s.f). Desarrollo web (Desarrollo Ágil Kanban). Recuperado de <http://www.desarrolloweb.com/articulos/desarrollo-agil-kanban.html>)

4.1. FASES DE LA EJECUCIÓN

4.1.1. Fase de inicio

En la fase de inicio de este proyecto se realizó el desarrollo de la investigación preliminar.

4.1.2. Fase de planeación

Durante esta fase se realizaron las actividades pertinentes al planeamiento del proyecto, inicialmente se realizó un análisis de la metodología que se utilizó en el desarrollo de la aplicación, a partir del estudio y comprensión de la metodología se planteó el cronograma de actividades incluido en el plan de proyecto. Dentro de esta etapa se realizó la preparación de los equipos de cómputo con la correspondiente instalación de las herramientas software para el desarrollo. En esta fase se realizó un proceso de aprendizaje de las herramientas a emplear.

En la fase de planeamiento también se llevó a cabo las reuniones respectivas entre el usuario final y los desarrolladores para el levantamiento de requerimientos y el desarrollo del modelo inicial de la aplicación. Con la especificación de requerimientos se inició la construcción de la lista de características para refinar de esta forma las funcionalidades que se incluyeron en la aplicación.

4.1.3. Fase de construcción

La fase de construcción abarco dos grandes actividades que tienen como característica principal su iteración, estas corresponden al diseño por característica y a construir por característica. Dentro del diseño está la identificación de clases, métodos, atributos, determinando las funciones que sean requeridas.

4.1.4. Fase de pruebas

Esta es la fase donde se revisó la consistencia de la aplicación, de acuerdo a un plan de pruebas adaptado a las necesidades de la aplicación y en concordancia con las directrices de la metodología. Se practicaron correcciones necesarias para el óptimo funcionamiento de la aplicación y se hace la entrega final de la misma.

5. FASE DE PLANEACIÓN

En el proceso de planeación se enfoca en primera instancia, que tan viable es el proyecto, en caso de una futura aplicación para negocio.

En esta etapa se desarrolló una encuesta en línea y se les pidió a 100 personas someterse a ella, con resultados positivos y el visto bueno para la creación de la apk.

Ya que básicamente todos los participantes de la encuesta poseen un teléfono inteligente, o sus familiares, y les pareció conveniente un aplicativo que les ayude con sus recordatorios médicos. Acá se obtuvieron los requerimientos según la necesidad del usuario.

5.1. REQUISITOS FUNCIONALES

- La aplicación debe permitir crear un usuario y guardar su información en la base de datos, este usuario puede tener varios perfiles dependiendo a cuanta personas le va a llevar el control de los recordatorios.
- La aplicación debe permitir actualizar y editar la información de los usuarios.
- La aplicación permitirá la creación y modificación de recordatorios en los módulos de citas médicas, exámenes médicos y sus requisitos e ingesta de medicamentos.
- La aplicación permitirá al usuario subir fotos de los productos médicos formulados si son necesarias en el módulo de medicamentos
- La aplicación dará aviso a la hora de recordar (el lugar, la hora, la concentración a tomar o los requerimientos necesarios para atender un evento que se gestiona con la agenda).
- La aplicación también enviara los recordatorios vía mensaje de texto cuando el usuario lo requiera.
- La aplicación permitirá evaluar al médico para tener un mejor conocimiento de este al usuario en un futuro.
- La aplicación tendrá una página web por la cual el usuario puede interactuar en ella sin necesidad del móvil.

5.2. REQUISITOS NO FUNCIONALES

- La aplicación se podrá visualizar en equipos Android incluso en los de gama baja.
- La página web se visualizará en cualquier navegador.
- La aplicación tendrá que visualizarse en un tiempo moderado.
- La aplicación debe seguir las reglas de seguridad de información de los usuarios.

5.3. CASOS DE USOS

Luego de tener los requerimientos, se crearon los casos de uso para tener una idea más centrada del desarrollo tanto de las bases de datos como de la interfaz de usuario.

5.3.1. Identificación de Actores

Figura 3 Identificación de Actores

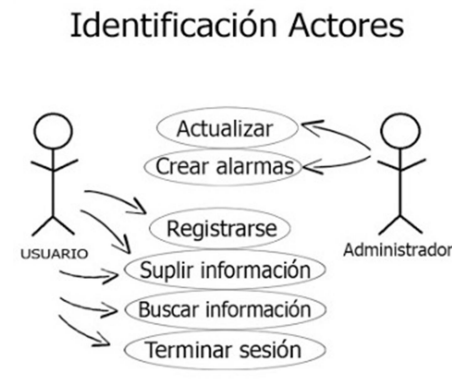


Tabla 1 Descripción del actor administrador

Actor:	Administrador
Casos de Uso Asociados:	Actualizar información, crear y gestionar alarmas
Descripción:	<p>El usuario Administrador será el encargado de mantener la aplicación a la vanguardia de los nuevos medicamentos, actualizando periódicamente las bases de datos, ya sea con las sugerencias, o bien con información de las páginas médicas.</p> <p>Además es el encargado de gestionar o verificar el correcto funcionamiento de la tabla q crea las alarmas de las notificaciones para el usuario</p>
Tipo:	Primario

Tabla 2 Descripción del actor usuario

Actor:	Usuario
Casos de Uso Asociados:	Registro, ingreso de información personal, calificación.
Descripción:	<p>El usuario promedio debe registrarse como en cualquier aplicación para poder acceder a la información, además de suplir lo correspondiente a su perfil para un mejor seguimiento a sus necesidades. Además debe calificar el servicio de sus doctores.</p>
Tipo:	Primario

5.3.2. Descripción Diagrama modelo casos de usos:

Figura 4 Diagrama Modelos de Casos de Usos

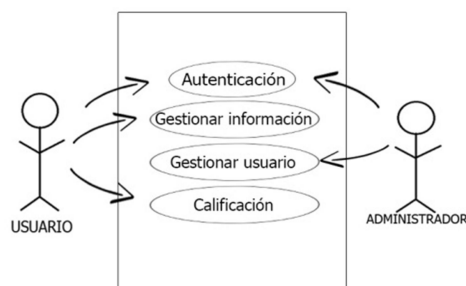


Tabla 3 Descripción Caso de Uso Autenticación

Caso De Uso:	Autenticación
Actores:	Administrador, usuario.
Propósito:	Permite al usuario o administrador ingresar y tener permisos de modificación.
Resumen:	Este caso de uso está diseñado para ser utilizado para administrar (en el usuario administrador) en la base de datos, y en el usuario ya en la apk, para tener acceso a la información.
Precondición:	✓ El usuario debe autenticarse
Flujo Principal:	El caso de uso se inicia cuando se muestra dos cajas de texto donde se debe ingresar los datos (usuario y contraseña), además de un botón de inicio de sesión que envía los datos para la autenticación.
Sub-flujo:	SF-1 Datos Se muestra dos cajas de texto para el ingreso de datos del usuario. En la primera caja se ingresa el nombre del usuario y en la segunda se ingresa la contraseña.
Excepciones:	EX-1 Debe especificar usuario, el campo usuario se encuentra vacío debe ingresar su nombre de usuario en el campo. EX-2 Debe especificar contraseña, el campo contraseña se encuentra vacío debe ingresar su contraseña en el campo EX-3 Verifique su usuario y contraseña, usuario o contraseña incorrectos, se solicita al usuario verificar e ingresar sus datos correctamente.

Tabla 4 Descripción Caso de Uso Gestionar Información

Caso De Uso:	Gestionar Información
Actores:	Usuario autenticado.
Propósito:	Permite al usuario, consultar, editar, eliminar y calificar sus citas, medicamentos, exámenes y doctores
Resumen:	En este caso de uso se permite extraer información de la base de datos para crear sus alarmas, además de adicionar información en su perfil, editarla eliminarla y en el caso de los doctores, calificar el servicio prestado para tenerlo presente a la hora de solicitar una nueva cita(pensado en que no todos quieren repetir malas experiencias)
Precondición:	El usuario debe desplazarse por cada uno de los formularios y generar la información que se almacenara para su posterior consulta
Flujo Principal:	El caso de uso se inicia cuando el usuario se ha autenticado correctamente. Acto seguido el usuario se puede desplazar por cada uno de los “módulos” de la aplicación ingresando su información para la posterior notificación o calificación. Además de la disposición del mismo para editar y eliminar
Sub-flujo:	SF-1 formularios, se muestran diferentes cajas de texto para recolectar la información del usuario.
Excepciones:	EX-1 cada una de las cajas tiene sus excepciones, ya sea de uso obligatorio de solo números de solo letras.

Tabla 5 Descripción Caso de Uso Gestionar Usuarios

Caso De Uso:	Gestionar Usuarios
Actores:	Administrador autenticado.
Propósito:	Permite al administrador mantener la aplicación a la vanguardia de los nuevos medicamentos y los nuevos exámenes, además de verificar el correcto funcionamiento de las notificaciones.
Resumen:	En este caso de uso el administrador debe ingresar información a la tabla de medicamentos y exámenes manteniendo actualizadas las mismas para beneficio de los usuarios.
Precondición:	✓ El administrador debe desplazarse por las tablas haciendo consultas updates e inserts.
Flujo Principal:	El caso de uso se inicia cuando el usuario administrador se ha autenticado correctamente. Acto seguido debe ir a cada una de las tablas a verificar la información modificarla o insertarla.
Sub-flujo:	SF-1 bases, cada columna necesita su formato específico.
Excepciones:	EX-1.

Tabla 6 Descripción Caso de Uso Calificación

Caso De Uso:	Calificación.
Actores:	Usuario autenticado.
Propósito:	Permite al usuario calificar los doctores previamente ingresados
Resumen:	En este caso de uso se permite calificar la experiencia del usuario con su doctor. Así se tendrá el registro para si se saca una nueva cita con el mismo o no.
Precondición:	✓ El usuario debe haber ingresado los datos del doctor previamente.
Flujo Principal:	El caso de uso se inicia cuando el usuario se ha autenticado correctamente. Acto seguido el usuario se desplaza al calificar doctor. Donde se mostraran los doctores que fueron previamente ingresados y se dará la opción de elegirlo para abrir un nuevo formulario donde calificar.
Sub-flujo:	SF-1 formulario, se muestran diferentes cajas de texto para recolectar la información de la experiencia del usuario.
Excepciones:	EX-1 cada una de las cajas tiene sus excepciones, ya sea de uso obligatorio de solo números de solo letras.

5.3.3. Descripción del Diagrama de caso de uso – Gestión información por usuario

Figura 5 Diagrama Gestión de Información

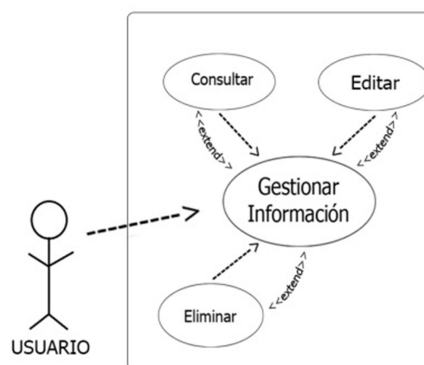


Tabla 7 Descripción Caso de Uso Consultar

Caso De Uso:	Consultar.
Actores:	Usuario autenticado.
Propósito:	Permite consultar la información deseada del usuario
Resumen:	Este caso de uso permite consultar la información que los usuarios han agregado con anterioridad
Precondición:	✓ El usuario debe estar registrado.
Flujo Principal:	El caso de uso se inicia cuando el usuario se ha autenticado correctamente. Acto seguido el usuario puede dirigirse a los módulos de consulta para mirar la información que se tiene.
Sub-flujo:	SF-1 el usuario debe elegir uno de sus registros para ver más información, editar o eliminar
Excepciones:	EX-1

Tabla 8 Descripción Caso de Uso Editar (usuario)

Caso De Uso:	Editar.
Actores:	Usuario autenticado.
Propósito:	Permite editar la información deseada del usuario
Resumen:	Este caso de uso permite editar la información que los usuarios han agregado con anterioridad
Precondición:	<ul style="list-style-type: none"> ✓ El usuario debe estar registrado. ✓ El usuario debe haber elegido el registro.
Flujo Principal:	El caso de uso se inicia cuando el usuario se ha autenticado correctamente. Acto seguido el usuario se dirige a los Módulos de información y elige un registro.
Sub-flujo:	SF-1 el usuario debe elegir uno de sus registros para poder editar la misma.
Excepciones:	EX-1 Debe dar clic al botón editar para acceder a la funcionalidad

Tabla 9 Descripción Caso de Uso Eliminar (usuario)

Caso De Uso:	Eliminar.
Actores:	Usuario autenticado.
Propósito:	Permite eliminar la información deseada del usuario
Resumen:	Este caso de uso permite eliminar la información que los usuarios han agregado con anterioridad
Precondición:	<ul style="list-style-type: none"> ✓ El usuario debe estar registrado. ✓ El usuario debe haber elegido el registro.
Flujo Principal:	El caso de uso se inicia cuando el usuario se ha autenticado correctamente. Acto seguido el usuario se dirige a los Módulos de información y elige un registro.
Sub-flujo:	SF-1 el usuario debe elegir uno de sus registros para poder elegir el botón eliminar.
Excepciones:	EX-1 Debe dar clic al botón eliminar para acceder a la funcionalidad

5.3.4. Diagrama de caso de uso – Gestión información usuarios

Figura 6 Diagrama Gestión de Usuarios

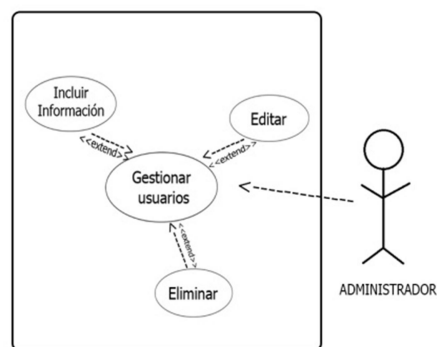


Tabla 10 Descripción Caso de Uso Consultar

Caso De Uso:	Consultar.
Actores:	Administrador autenticado.
Propósito:	Permite consultar la información deseada de las bases de datos.
Resumen:	Este caso de uso permite consultar la información en las bases de datos de las tablas de medicamentos y exámenes
Precondición:	✓ El administrador debe estar registrado.
Flujo Principal:	El caso de uso se inicia cuando el administrador se ha autenticado correctamente. Acto seguido puede dirigirse a la base de datos y elegir la tabla a consultar
Sub-flujo:	SF-1 el administrador debe tener presente la información y tablas que desea consultar
Excepciones:	EX1

Tabla 11 Descripción Caso de Uso Editar (administrador)

Caso De Uso:	Editar.
Actores:	Administrador autenticado.
Propósito:	Permite editar la información deseada de las tablas
Resumen:	Este caso de uso permite editar la información
Precondición:	<ul style="list-style-type: none"> ✓ El administrador debe estar registrado. ✓ El administrador debe haber elegido el registro.
Flujo Principal:	El caso de uso se inicia cuando el administrador se ha autenticado correctamente. Acto seguido el administrador debe hacer el update correspondiente
Sub-flujo:	SF-1 el administrador debe elegir uno de los registros para poder editar la misma.
Excepciones:	EX1

Tabla 12 Descripción Caso de Uso Eliminar (administrador)

Caso De Uso:	Eliminar.
Actores:	Administrador autenticado.
Propósito:	Permite eliminar la información deseada del usuario
Resumen:	Este caso de uso permite eliminar la información que los usuarios han agregado con anterioridad
Precondición:	<ul style="list-style-type: none"> ✓ El administrador debe estar registrado. ✓ El administrador debe haber elegido el registro.
Flujo Principal:	El caso de uso se inicia cuando el administrador se ha autenticado correctamente. Acto seguido el administrador debe hacer el delete correspondiente
Sub-flujo:	SF-1 el usuario debe elegir uno de sus registros para poder elegir el botón eliminar.
Excepciones:	EX-1

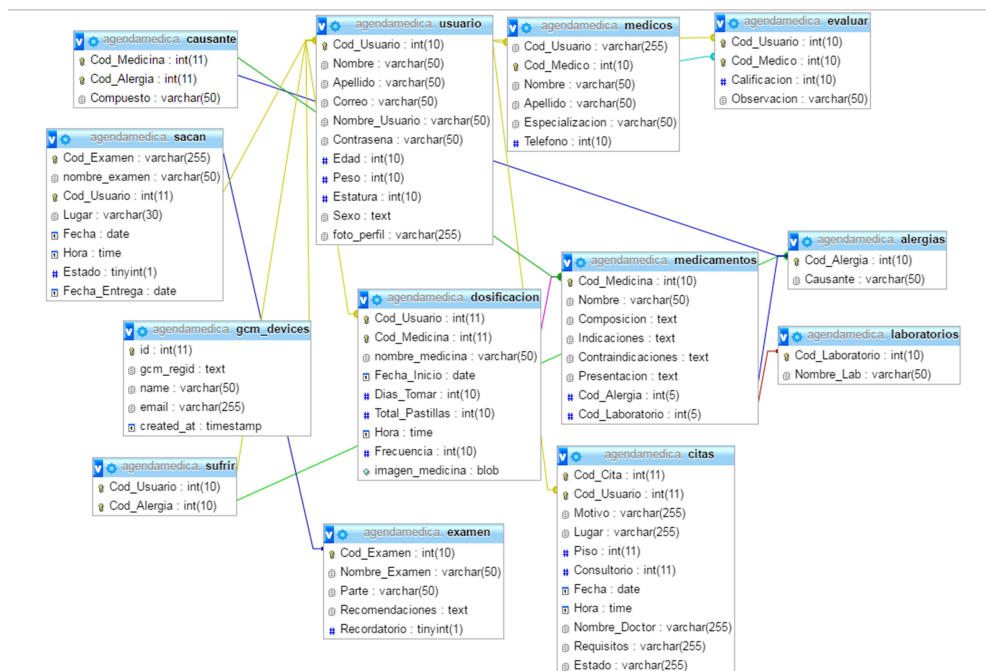
6. FASE DE CONSTRUCCIÓN

Se empezó a deliberar sobre la base de datos, a medida que se planteaba la interfaz del usuario, tratando en medida de hacerla lo más amigable posible, para que así al usarla fuese lo más simple posible.

6.1. BASE DE DATOS

La base de datos fue desarrollada en SQL con el gestor phpmysqladmin creada en el entorno gráfico con sus respectivas relaciones dependiendo de qué información sea necesaria por parte del usuario o que necesita el usuario por parte del administrador. Para la elaboración de la base de datos se determinaron las siguientes entidades iniciales como: usuario, medicamentos, sacan (exámenes), citas, médicos y evaluar; como es una base de datos relacional cada tabla está vinculada de una u otra forma con el usuario principal. Las tablas de medicinas se relacionan con las alergias y las de sacan se relacionan con los exámenes. A continuación se presenta el modelo entidad/relación de la base de datos.

Figura 7 Modelo Entidad Relación de la Base de Datos

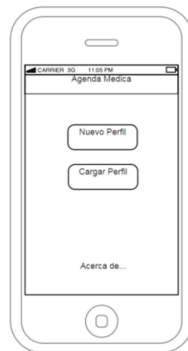


6.2. MOCKUPS DE LA APLICACIÓN

Los Mockups que se presentaran a continuación. Son los que al final quedaron en la aplicación final.

Inicio: la aplicación se disponen tres opciones una para ver la información sobre las desarrolladoras de la aplicación y las otras dos son la opción de registro e inicio de sesión las cuales desplegaran a su vez un formulario de registro o un formulario de inicio de sesión.

Figura 8 Mockup Página de Inicio



Formulario de Registro: en este formulario vienen las preguntas básicas además de personales para ir creando su perfil en las bases de datos. Puede elegir las alergias a los medicamentos que conoce con otro subformulario al registrarse volverá a la pantalla anterior para que inicie sesión.

Figura 9 Mockup Formulario de Registro



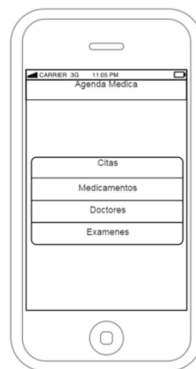
Inicio de Sesión: Como esta es una aplicación multipersonal al iniciar la sesión se presentaran los perfiles que tiene a su disposición en esta pantalla solo se pedirá el nombre de usuario y la contraseña previamente indicadas, al seleccionar alguno puede o bien regresar o eliminar o continuar al menú principal.

Figura 10 Mockup Inicio de Sesión



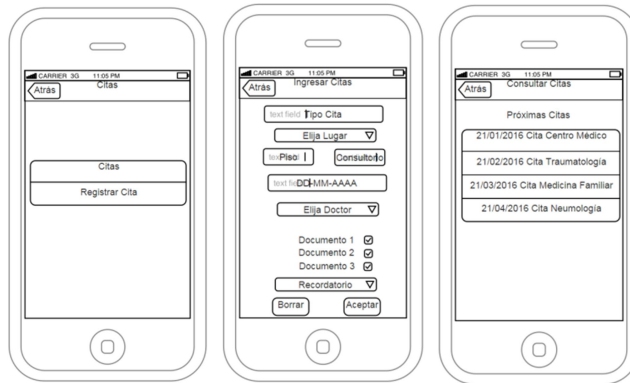
Menu Principal: En el menu principal nos encontraremos con las opciones que tenemos a disposicion:

Figura 11 Mockup Menú Principal



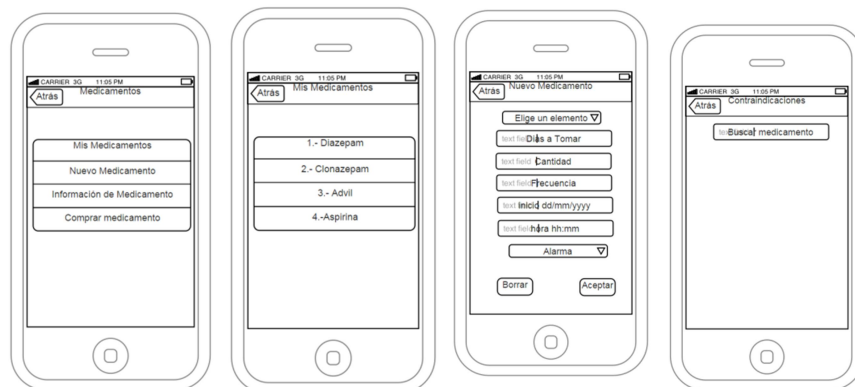
Citas: esta opción tiene dos opciones que abren sus respectivas pantallas, mirar las citas que ha registrado y agregar una nueva cita.

Figura 12 Mockup Módulo Citas



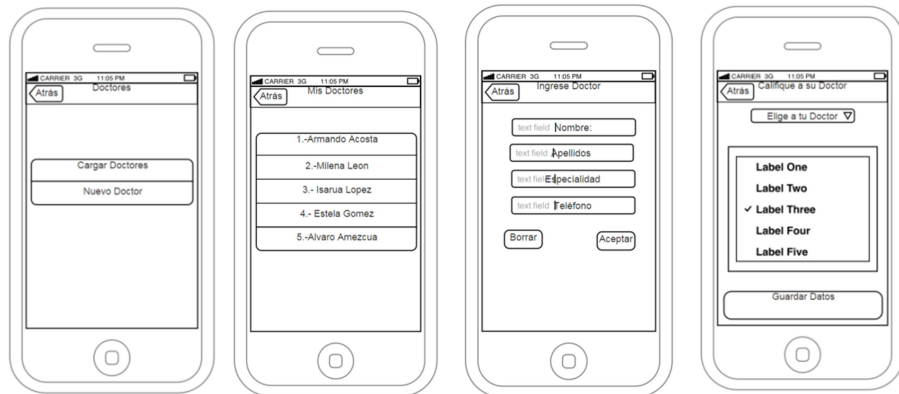
Medicamentos: esta opción de menú abre 4 nuevas opciones ver los medicamentos en mi usuario agregar un medicamento. Que a su vez abre un nuevo formulario y un consultor para ver las indicaciones contraindicaciones y demás información presente en las bases de datos sobre este medicamento.

Figura 13 Mockup Módulo Medicamentos



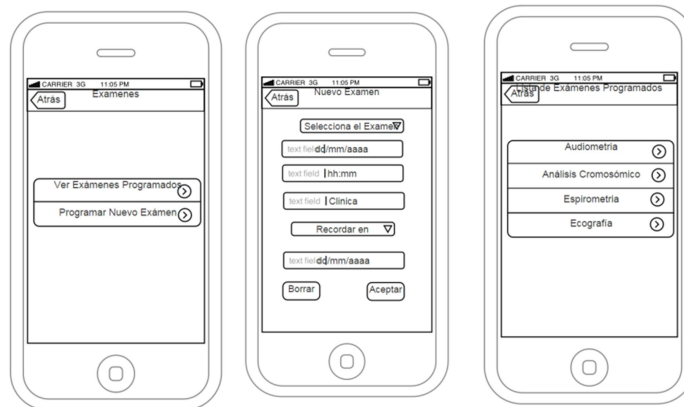
Médicos: este módulo tiene tres opciones: agregar médico, ver los médicos y además poder calificarlos según la forma de atención.

Figura 14 Mockup Módulo Médicos



Exámenes: esta opción tiene dos opciones que abren sus respectivas pantallas, mirar los exámenes que ha registrado y agregar una nuevo examen.

Figura 15 Mockup Módulo Exámenes



Resultados alcanzados:

Los resultados fueron positivos, dando en primera instancia el aval por parte del usuario al proyecto y además dando como resultado una interfaz y un modelo de base de datos funcional y útil para los usuarios, con la creación de los casos de uso se pudo ver cada una de las funcionalidades que se le darán al proyecto.

Dificultades:

Una de las dificultades que se presentaron al iniciar el desarrollo fue elegir entre tantas opciones de motores de bases de datos y de frameworks, ya que no solo se busca facilidad de usabilidad, sino se debe estar seguro de la cantidad de documentación que se encontrará para poder resolver cualquier inconveniente, y dado que se estaba frente a una aplicación móvil, la documentación de los errores presenciados muchas veces no era suficiente.

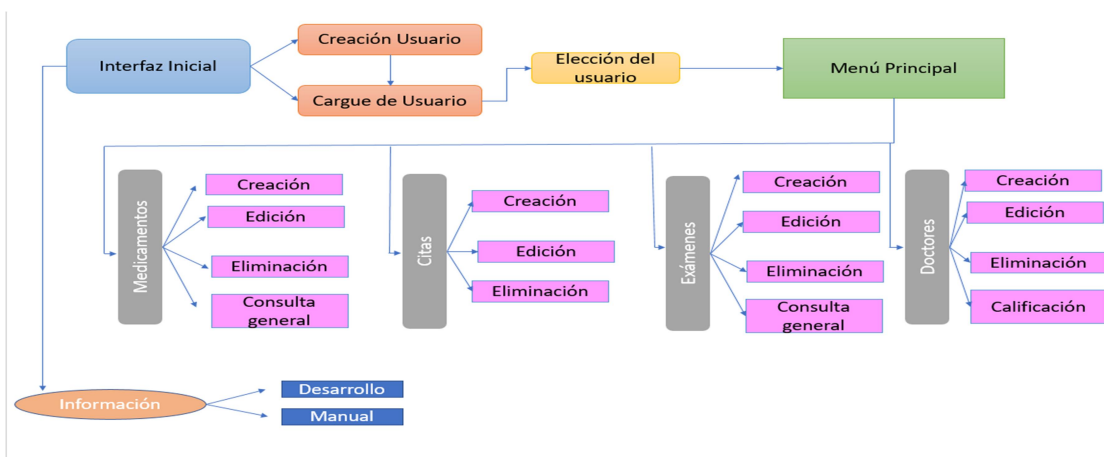
También se presentaron dificultades en cuanto a la migración de móvil a web, muchas de las librerías hacían conflicto en el navegador, por lo que fue casi necesario programar de nuevo, además que las constantes actualizaciones de las librerías, de apache cordova y Android fueron una piedra de tropiezo a la hora de compilar y agregar nuevas funcionalidades ya que pedían constantes actualizaciones que en algunas ocasiones presentaban inconvenientes con la app.

Como en un inicio se usó Netbeans para crear la app nativa los SDK consumían mucho y esto representaba un problema de tiempo además que los emuladores no corrían en cualquier equipo cosa que represento una dificultad a la hora de las pruebas de la app.

6.3. IMPLEMENTACIÓN

En la figura 16 se muestra cómo que va a encontrar el usuario a la hora de interactuar con la aplicación móvil y web ya que ambas son similares en sus módulos.

Figura 16 Mapa mental sobre el uso de la aplicación



Lo anterior se desarrolló gracias a que se siguió el patrón modelo, vista, controlador (MVC), por lo cual se generó una aplicación ágil y liviana que correrá en cualquier navegador y cualquier versión Android.

6.3.1. Componente modelo

A continuación se presenta un ejemplo de un modelo que se desarrolló para la aplicación:

Figura 17 Ejemplo código para un modelo

```
1 <?php
2 include("Coneccion.php");
3
4 $usuario = $_POST['usuario'];
5 $validar = $_POST['validar'];
6 $motivo = $_POST['motivo'];
7 $lugar = $_POST['lugar'];
8 $piso = $_POST['piso'];
9 $consultorio = $_POST['consultorio'];
10 $fecha = $_POST['fecha'];
11 $hora = $_POST['hora'];
12 $doctor = $_POST['doctor'];
13 $requisitos = $_POST['requisitos'];
14 $estado = $_POST['estado'];
15
16 if($validar == 0){
17     $consulta = "INSERT INTO citas ( Cod_Usuario, Motivo, Lugar, Piso, Consultorio, Fecha, Hora, Nombre_Doctor, Requisitos, Estado ) VALUES ( '
18         usuario', '$motivo', '$lugar', '$piso', '$consultorio', '$fecha', '$hora', '$doctor', '$requisitos', '$estado' )";
19     if( mysql_query ($consulta, $conex)){
20         echo "0";
21     }else{
22         echo "1";
23     }
24 }else{
25     $consulta = "UPDATE citas SET Cod_Usuario= '$usuario', Motivo ='$motivo', Lugar='$lugar', Piso='$piso', Consultorio='$consultorio', Fecha='$
26     fecha', Hora='$hora', Nombre_Doctor='$doctor', Requisitos='$requisitos', Estado ='$estado' WHERE Cod_Usuario= '$usuario' AND Motivo ='$
27     motivo' ";
28     if( mysql_query ($consulta, $conex)){
29         echo "0";
30     }else{
31         echo "1";
32     }
33 }
```

En caso de la aplicación los modelos son varios y están desarrollados en php alojados en el servidor. Cuando el controlador interpreta la entrada de un usuario accede al modelo y lo actualiza según la solicitud del usuario.

6.3.2. Componente vista

Se presenta a continuación un ejemplo de una vista que se desarrolló para la aplicación.

Figura 18 Ejemplo del código vista

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4
5 <!-- Basic Page Needs
6 =====>
7 <meta name="format-detection" content="telephone=no">
8 <meta name="application-tap-highlight" content="no">
9 <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width">
10 <meta content="text/html; charset=utf-8">
11 <!--<meta http-equiv="Content-Security-Policy" content="default-src *; style-src 'self' http://* 'unsafe-inline'; script-src 'self'
12 http://* 'unsafe-inline' 'unsafe-eval'"/>
13 <link href="https://code.jquery.com/jquery-1.4.5/jquery.mobile-1.4.5.css" rel="stylesheet" type="text/css"/>
14 <title>Agenda Medica : Agregar Citas</title>
15 <link rel="stylesheet" href="fonts.css">
16
17 <!-- Mobile Specific Metas
18 =====>
19 <!-- Favicon -->
20 <link rel="shortcut icon" type="image/icon" href="images/favicon.ico"/>
21
22 <!-- CSS
23 =====>
24 <!-- Bootstrap css file-->
25 <link href="css/bootstrap.min.css" rel="stylesheet">
26 <!-- Font awesome css file-->
27 <link href="css/font-awesome.min.css" rel="stylesheet">
28 <!-- Default theme css file -->
29 <link id="switcher" href="css/themes/default-theme.css" rel="stylesheet">
30 <!-- Slick slider css file -->
31 <link href="css/slick.css" rel="stylesheet">
32
33 <!-- Google fonts -->
34 <link href="http://fonts.googleapis.com/css?family=Raleway" rel="stylesheet" type="text/css">
35 <link href="http://fonts.googleapis.com/css?family=Habibi" rel="stylesheet" type="text/css">
36 <link href="http://fonts.googleapis.com/css?family=Cinzel+Decorative:900" rel="stylesheet" type="text/css">
37
38 <link rel="stylesheet" href="css/estilos.css">
39
40 <script src="http://code.jquery.com/jquery-latest.js"></script>
41 <script src="js/jk/main.js"></script>
42 <script src="js/jk/background.adaptable.js"></script>
43 <script type="text/javascript">
44
45 $(document).ready(function () {
46
47     var paramstr = window.location.search.substr(1);
48     var paramarr = paramstr.split("&");
49     var params = {};
50
51     for ( var i = 0; i < paramarr.length; i++) {
52         var tparam = paramarr[i].split("=");
53         params[tparam[0]] = tparam[1];
54     }
55
56     $("#id_user").val(params['id_user']);
57 </script>
58
59 <style type="text/css">
60
61 table {
62     margin-top: 50px;
63     font: 75%/1.5em arial, geneva, sans-serif;
64     border-collapse: initial;
65 }
66
67
68 td {
69     border-bottom: 20px solid #fff;
70     border-left: 20px dotted #fff;
71     padding: 30px;
72     border-style: outset;
73 }
74
75 tr {
76     background: #FFFFFF;
77 }
78
79 body {
80     background-image: url(images/universe.jpg);
81     background-size: cover;
82 }
83 </style>
84
85 </head>
86 <body>
87 <!-- id del usuario loggeado -->
88 <input id="id_user" style="display:none;" value="">
89 <input id="id_medimento" style="display:none;" value="">
90 <input id="id_cita" style="display:none;" value="">
91 <input id="id_examen" style="display:none;" value="">
92 <input id="id_doctor" style="display:none;" value="">
93
94 <!-- BEGAIN PRELOADER -->
95 <div id="preloader">
96 <div id="status"&nbsp;&nbsp;&nbsp;</div>
97 </div>
98 <!-- END PRELOADER -->
99
```

Se utiliza HTML para poder visualizar con sencillez la información necesaria de la aplicación. El controlador le envía a la vista la tarea de desplegar la interfaz de

usuario y la vista obtiene los datos del modelo y así genera como respuesta la interfaz apropiada al usuario donde se reflejara los cambios en el modelo.

6.3.3. Componente controlador

Se presenta a continuación un ejemplo de un controlador que se desarrolló para la aplicación:

Figura 19 Ejemplo código de un controlador

```
142 $.getJSON('http://190.144.3.210:88/2agendamedica/obtener_examen.php?id_user='+id_user, function(data){
143     console.log(JSON.stringify(data));
144     var datos;
145     $("#lista_mis_examenes").empty();
146     var contador=0;
147     if (data.length ==0) {
148         $("#lista_mis_citas").append(
149             "<div > <strong> No hay registros en nuestras Bases de Datos!, Agrega un Examen! </strong></div> <div>&nbsp;</div>"
150         );
151     }
152     $(data).each(function (index,data) {
153         contador++;
154         $("#lista_mis_examenes").append(
155             "<div class='row padding lista_examen' id='lista_examen_"+contador+"' onclick='css_selec_examen(this.id,\""+data.Cod_Examen+"');"> "
156             "<div class='col-xs-12'>"+
157                 "<strong>Examen: "+contador+"Codigo: "+data.Cod_Examen+"</strong>"+
158             "</div>"+
159             "<div class='col-xs-12'>"+
160                 "<strong>Examen: </strong>"+data.Nom_Examen+" "+
161             "</div>"+
162             "<div class='col-xs-12'>"+
163                 "<strong>Lugar: </strong>"+data.Lugar+" "+
164             "</div>"+
165             "<div class='col-xs-12'>"+
166                 "<strong>Fecha Examen: </strong>"+data.Fecha+" "+
167             "</div>"+
168             "<div class='col-xs-12'>"+
169                 "<strong>Hora: </strong>"+data.Hora+" "+
170             "</div>"+
171             "<div class='col-xs-12'>"+
172                 "<strong>Estado: </strong>"+data.Estado+" "+
173             "</div>"+
174             "<div class='col-xs-12'>"+
175                 "<strong>Fecha de entrega: </strong>"+data.Fecha_Entrega+" "+
176             "</div>"+ "<div> &nbsp;</div> "+
177         "</div>"
178     );
179     });
180     .fail(function() {
181         console.log( "error" );
182         //alert('No hay conexion');
183         $("#div_error_conexion").show();
184     });
185     });
186     //ajax para obtener mis citas
187     function obtener_mis_citas(){
188     191
```

Desarrollada en JavaScript, jQuery y Ajax. El controlador recibe las notificaciones de las acciones del usuario, este gestiona los eventos por medio de un gestor de eventos, trae la información y complementa las vistas, a su vez la vista accede al controlador por medio de oncliks.

6.3.4. APIs Usados.

Ya que hay APIs creadas se les dio de uso para crear funcionalidades para la aplicación, tales como el envío de notificaciones al celular (usando el registro del móvil para acceder a él), y el envío de mensaje de texto (funcionalidad opcional que envía SMS al celular proporcionado por el usuario).

Figura 20 Código para acceder a la API de envío de notificaciones push

```
1 <?php
2 class GCM {
3     function __construct() {
4     }
5
6
7     /*--- Enviando notificaciones push ----*/
8     public function send_notification($registatoin_ids, $message) {
9         include_once 'config.php';
10        // variable post http://developer.android.com/google/gcm/http.html#auth
11        $url = 'https://android.googleapis.com/gcm/send';
12
13        $fields = array(
14            'registration_ids' => $registatoin_ids,
15            'data' => $message,
16        );
17
18        $headers = array(
19            'Authorization: key=' . GOOGLE_API_KEY,
20            'Content-Type: application/json'
21        );
22        // abriendo la conexion
23        $ch = curl_init();
24
25        // Set the url, number of POST vars, POST data
26        curl_setopt($ch, CURLOPT_URL, $url);
27
28        curl_setopt($ch, CURLOPT_POST, true);
29        curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
30        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
31
32        // Deshabilitamos soporte de certificado SSL temporalmente
33        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
34
35        curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($fields));
36
37        // ejecutamos el post
38        $result = curl_exec($ch);
39        if ($result == FALSE) {
40            die('Curl failed: ' . curl_error($ch));
41        }
42
43        // Cerramos la conexion
44        curl_close($ch);
45        echo $result;
46    }
47
48 }
```

Por medio del cURL se puede acceder a estas APIs y se les transfiere la información que es requerida en la web que aloja la funcionalidad.²⁰De esta forma se pueden crear los objetivos descritos con anterioridad.

²⁰ Google Cloud Messagin, (s.f). Google Apis. Recuperado de <https://android.googleapis.com/gcm/send>

Tú lo envías. (s.f) APIS. Recuperado de <http://api.tuloenvias.com/sms/>

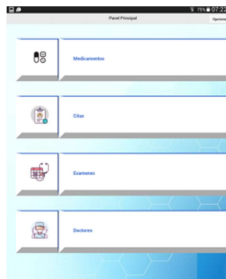
6.4. RESULTADOS DE LA IMPLEMENTACIÓN DEL PROTOTIPO

Al final de la implementación se tiene un prototipo cuya meta era la plataforma Android pero que con el pasar del tiempo y el estudio de muchas vías de programación se desarrolló una aplicación móvil y una página web que puede ejecutarse en móviles incluso de poca potencia y en diferentes navegadores consecutivamente, también posee soporte en la nube; que ayudara a las personas con un recordatorio y guardara información importante a la cual podrán acceder en cualquier momento que lo soliciten mediante una autenticación.

6.4.1. Versión Móvil

Fue un trabajo desarrollado en PhoneGap bajo los lenguajes HTML, JavaScript, JQueryMobile y Ajax, usando librerías de cordova para el correcto uso de batería, alarmas, alertas, notificaciones y uso de la cámara así como de otros recursos del móvil que dio como resultado una aplicación amigable, que debido a las pocas líneas compiladas en el lado del usuario es muy portable y genera muy pocos recursos gastados. Esto debido a que la mayoría de información es solicitada y enviada a los servidores en la nube por medio de peticiones php cuyo desarrollo se encuentran alojadas en las mismas, esto da como resultado una aplicación que necesita wifi o bien datos. Ya que basados en los requerimientos levantados en las encuestas iniciales se pudo sacar un producto final. Aquello que los usuarios deseaban, una agenda médica que no solo le recordará cuando tomarse una pastilla, sino que por medio de la visualización de imágenes, le dará mayor información en aquellos casos en los que las formulaciones sean extensas, y le hará control a sus citas, exámenes y doctores, esto prestándole especial atención a su salud y si así lo desea a la salud de su núcleo familiar. Esto dado a que cada vez que registre un usuario con su móvil el id de registro será el mismo y así tendrá se tendrá más control sobre la salud de un determinado grupo.

Figura 21 Pantallazo de la aplicación móvil (Modulo medicamentos)



6.4.2. Versión Web.

El uso de las tecnologías de desarrollo en phonegap fue un gran avance para la realización web del sitio de agenda médica, si bien hubo muchos cambios entre una versión y la otra y la versión web tiene más líneas de código y un poco más funcionalidades que la aplicación móvil. Esta versión también es ligera y muy fácil de usar, con un cómodo menú adaptable que no tiene pierde ya que está muy bien especificado. Esta aplicación les es muy útil a los usuarios que la usan ya sea para visualizar la información diligenciada en la app como para diligenciar los formularios o bien eliminar o editarlos en la comodidad de su computador. Cabe resaltar que desde esta aplicación no se harán registros, esto porque en el registro es necesario el id de registro del móvil para el correcto funcionamiento del envío de las notificaciones, este sitio web está alojado en un servidor 100% funcional y el cual cuenta con dos versiones de visualización, la versión simple que tiene la misma interfaz gráfica de la versión móvil y que es más liviana en cuanto al consumo de recursos en internet, y la versión normal, que no es solo más agradable visualmente sino que presenta mayor información a quien lo necesite.

6.4.3. Implementación Base de Datos

Con un motor MySQL se creó en los servidores una base de datos que no solo guardara la información del usuario final, sino que está poblada de información pertinente para estos, una serie de información que le facilitara la vida a los usuarios debido a la vasta información que se tiene de los medicamentos y diferentes tipos de exámenes que se pueden enviar.

Esta base de datos está directamente relacionada entre sí para brindar mayor usabilidad de la información de la misma, además está conectada a las diferentes versiones de la aplicación lo cual hará que la información agregada en un lado sea visible en el otro, accediendo a ella por medio de consultas SQL en los modelos presentados en cada uno de los archivos .php.

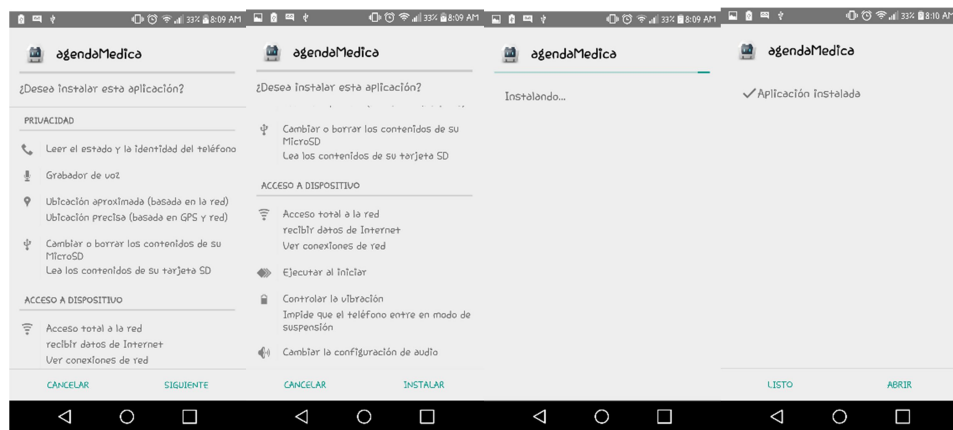
La compilación final de estos elementos dio como resultado una buena aplicación que cualquiera puede llevar en su bolsillo y acceder desde cualquier navegador con sus credenciales, esto le ayudara a recordar toda la información pertinente y en caso de que no tenga acceso a internet, si el usuario lo desea recibirá mensajes de texto.

6.4.4. Implementación apk

La aplicación está en un práctico apk que si bien no se ha podido cargar en la playstore para que cualquier usuario pueda descargarla por cuestiones de recursos que son solicitados por google play. Pero en las fases de pruebas se solicitó a cada usuario su teléfono y por medio de la conexión de transferencia de archivos se pasa el apk.

Acto seguido el usuario podrá abrirlo en su celular e instalarlo.

Figura 22 Pantallazo Instalacion del apk

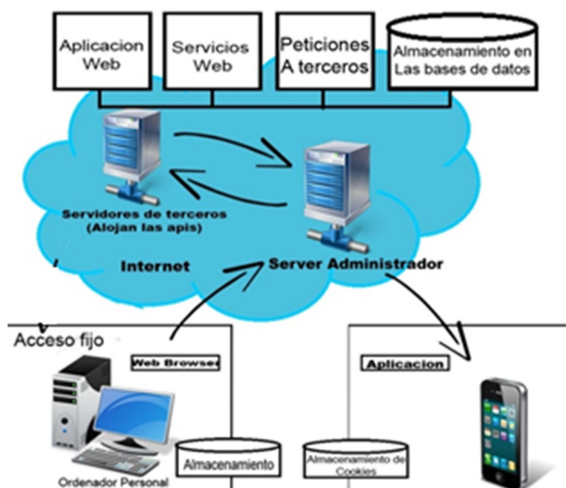


Luego de la instalación la aplicación queda lista para ser usada y entra al modo de usuarios bien sea para ingresar a un usuario ya creado, o para crear un nuevo usuario.

En la pestaña de informacion se podra encontrar una opcion de ayuda que presenta un pequeño manual de usuario que mostraran los pasos a seguir en caso de perdida dentro de la aplicación.

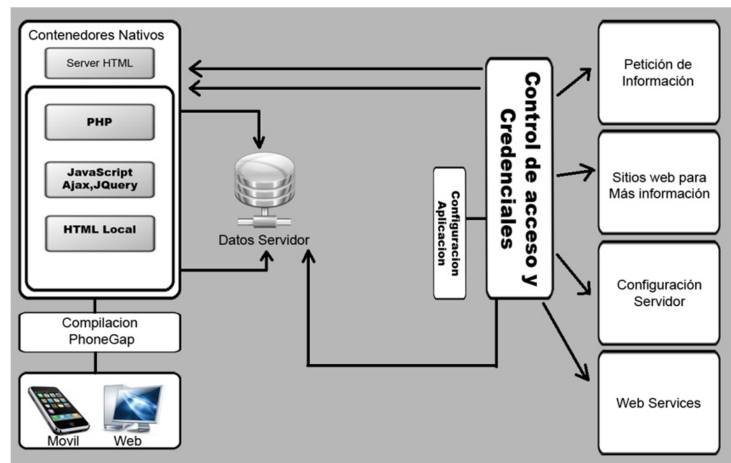
6.5. ARQUITECTURA

Figura 23 Arquitectura del despliegue de la página web y de la aplicación móvil.



La solución se abordó de modos diferentes; en lo nativo se elaboraron dos formas de interacción con el servidor principal, uno que se visualiza a través de un browser y el otro a través de la aplicación móvil. Ambos accesos guardan información de tipo local un poco limitada ya que necesitan internet para trabajar con la información alojada, la cual es solicitada a través del servidor en la nube, este server a su vez cuando se hace una petición (ya sea a terceros), a sus servicios web o bien a la base de datos, que no solo recibe información sino que de forma recíproca solicita al servidor una petición (función del servidor que constantemente solicita información a la base de datos) a servidores terceros los cuales alojan las APIs que a través de un servicio web enviará notificaciones push y sms al móvil. Los contenedores nativos para ambos modos de acceso están basados en los mismos lenguajes de programación HTML, php, JavaScript, JQuery y Ajax, como ya se ha hablado con antelación y compilados de formas diferentes, el uno con phoneGap o cordovabuild y el otro alojado en el servidor para acceder a él desde un browser. Ambos acceden a la información por medio de los controles de acceso que tiene una serie de configuraciones local y en el servidor alojado; a partir de este momento se puede acceder a los servicios de la aplicación móvil y/o su aplicación web.

Figura 24 Diagrama de la arquitectura la página web y de la aplicación móvil.



El flujo que se siguen los usuarios es sencillo de entender, los usuarios solo deben proporcionar la información a llevar a la base de datos, eliminarla o editarla y en caso de los exámenes y medicamentos hacer las peticiones para obtener los datos que se requieren de los diferentes listados que están poblados en la base de datos y en cuanto al doctor el calificarlos con un formulario más. Lo otro lo hace la aplicación: las notificaciones y mensajes de texto son peticiones que hace el server a los terceros.

7. FASE DE PRUEBAS

7.1. PRUEBAS DE USABILIDAD

Se les aplicó un breve cuestionario a diferentes usuarios. Este cuestionario dio como resultado que los usuarios tuvieron pocos problemas a la hora de usar la aplicación.

7.2. PASOS QUE SE SIGUIERON

- Se creó un cuestionario donde calificaran la usabilidad de la aplicación.
- Se asignaron diferentes actividades que el usuario realizaba en la aplicación.
- Se invitó a varios usuarios a realizar la prueba.
- Se realizaron las pruebas.
- Se analizaron los datos obtenidos.

Usuarios seleccionados: Fueron personas que son clientes de una droguería y que aceptaron realizar la prueba.

Creación del cuestionario: Se les explico a los usuarios como funciona básicamente la aplicación y luego se le dieron diferentes tareas a realizar.

Tareas a realizar:

- Crear un usuario nuevo ingresando los datos solicitados.
- Ingresar con el usuario anteriormente creado con clave incorrecta y luego con la correcta.
- Crear un recordatorio de cita, de exámenes y uno de ingesta de medicamentos (incluida la foto del medicamento seleccionado)
- Notificar si recibió los recordatorios a la hora indicada.
- Notificar si recibió un mensaje de texto con los recordatorios programados a la hora indicada.

8. CONCLUSIONES

El uso del patrón MVC (modelo, vista, controlador) fue una ventaja ya que es una técnica muy práctica en el desarrollo de tecnologías que necesariamente deben tener un flujo liviano, ya que ayudara a que la mayor cantidad de código sea alojado en el servidor.

Se pudo concluir que el uso del software para el correcto desarrollo de aplicaciones como lo es PhoneGap en este caso es muy útil a la hora de probar la aplicación de forma local y de compilar las apks.

Debido a la extensa información que se encuentra en Internet sobre la creación de aplicaciones en el idioma de inglés mayormente, se llegó a la premisa de que el conocimiento aunque sea básico de este idioma es importante para cualquier ingeniero, por lo tanto es de suma importancia aprenderlo.

Dado que phoneGap está basado en programación web el conocimiento en los lenguajes de desarrollo (HTML, JavaScript y CSS) es de suma importancia y de gran ayuda; por esto debido a la sólida y vasta información que se pudo obtener en la Universidad, libros e Internet, se logró el desarrollo de esta aplicación móvil y se obtuvieron grandes conocimientos a la hora de crear páginas web, *web services* y todo tipo de interacciones para los usuarios finales en aplicaciones web.

Esta aplicación se puede idear para que en un futuro cercano tenga la opción de presentarse en una convocatoria del Ministerio de Tecnologías de la Información y las Comunicaciones de Colombia, partiendo de la gran base con la que se cuenta para mejorarla y de este modo obtener un producto que se pueda vender ya sea por publicidad mostrada en la misma o por versiones de la misma que serán pagas. Además al hablar con los sectores de salud se encontró que algunas droguerías están dispuestas a obtener el software para que los usuarios puedan obtener con facilidad sus productos médicos. Y una IPS del servicio oncológico y sus diferentes sedes del país, también está interesada en el software para recordarle al usuario sus citas de quimioterapia y la ingesta de medicamentos en sus respectivos tratamientos contra el cáncer. Por lo tanto esta aplicación puede dar origen a un modelo de negocio muy fuerte.

En conclusión gracias a este proyecto de investigación se pudo interactuar con los usuarios. Levantar los requerimientos, diseñar la arquitectura y el desarrollo del software; se aprendió durante el proceso diferentes lenguajes de programación (HTML, ajax, php, jQuery) que ayudara a los autores en el mundo laboral con la experiencia recolectada en este proceso.

9. RECOMENDACIONES

Para futuras implementaciones del prototipo aquí presentado se recomienda tener en primer lugar un buen hardware para pruebas. No bastarse solo con la máquina virtual que poseen las herramientas de desarrollo móvil, se debe tener unas buenas bases en programación móvil, además de saber elegir la herramienta que mejor se adapte a las necesidades, capacidades, y aptitudes del desarrollador.

Aunque en Internet hay varios cursos que pueden ayudar a lograr el objetivo de esta implementación se debe esforzar en cuanto a la búsqueda de información que le ayude a lograr el objetivo ya sea para una sola plataforma o sea multiplataforma.

Se espera que el desarrollador tenga fuertes bases en bases de datos a la hora de querer ingresar más información o implementarle más cosas a la aplicación, además de tener claros los conceptos de programación en la nube.

10. BIBLIOGRAFÍA

- Organización Mundial de la Salud. (s.f). Salud mental (Definición). Recuperado de http://www.who.int/topics/mental_health/es/
- Colegio de Farmacéuticos de la Provincia de Barcelona. (s.f). Subdepartamento de Registro (Definición de Medicamentos). Recuperado de http://www.ispch.cl/anamed/subdeptoregistro/definicion_medicamentos
- COLOMBIA. CONGRESO DE LA REPÚBLICA. Ley 100. (23, diciembre, 1993). Por la cual se crea el sistema de seguridad social integral y se dictan otras disposiciones. Diario Oficial. Bogotá, D.C., 1993. no. 41148. p. 1-168.
- Ministerio de salud. Listados actualizados de medicamentos, procedimientos, y servicios incluidos tanto en el POS del régimen contributivo como en el POS del régimen subsidiado(Sitio web del Ministerio de Salud y Protección Social. <https://www.minsalud.gov.co/salud/Paginas/pos.aspx>
- Flanagan, David. JavaScript; The Definitive Guide (Gravenstein Highway North, O'Reilly Media, 2011). Pag:1-3
- Fritzler, Danilo. (2006). Desarrollo Web (JavaScript a fondo). Recuperado de <http://www.desarrolloweb.com/javascript/#librerias>
- Nixon, Robin. Learning PHP, MySQL, JavaScript, CSS & HTMML5 (Gravenstein Highway North, O'Reilly Media, 2014). PÁG:323-328
- Dayley, Brad. JQuery and JavaScript. (U.S, Addison-Wesley, 2014) PÁG: 12-14.
- Álvarez, Miguel Ángel. (s.f). Desarrollo web (Manual de PHP). Recuperado de <http://www.desarrolloweb.com/manuales/manual-php.html#capitulos12>
- Rodríguez Rancel, Mario. (s.f). Aprender a programar (AJAX). Recuperado de http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=882:i-que-es-y-para-que-sirve-ajax-ventajas-e-inconvenientes-

javascript-asincrono-xml-y-json-cu01193e&catid=78:tutorial-basico-programador-web-javascript-desde-&Itemid=206

- Romero García, Israel. (2013). Acerca de HTML (Manual de HTML). Recuperado de <http://www.acercadehtml.com/manual-html.html>
- Kyrnin, Jennifer. HTML5 Mobile Application Development in 24 Hours. (U.S, SAMS).
- Fritzier, Danilo. (2008). Culturación (Que es y Para qué Sirve MySQL). Recuperado de <http://culturacion.com/que-es-y-para-que-sirve-mysql/>
- Universidad de Concepción. (s.f). Common Gateway Interface CGI. Recuperado de <http://www.inf.udec.cl/~jlopez/DSWR/cgi.html>
- Van Der Henst Solis, Christian. (2001). Maestros Del Web (¿Qué es el CGI?). Recuperado de <http://www.maestrosdelweb.com/cgiintro/>
- Cordova (Documentación Apache Cordova). (s.f.). Recuperado de <https://cordova.apache.org/docs/es/latest/guide/overview/>
- Wargo, John M. APACHE CORDOVA 3 PROGRAMMING (U.S, Addison-Wesley, 2013) PÁG:5-14
- Pérez Esteso, Mario. (s.f). GeekyTheory (Programación extrema). Recuperado de <https://geekytheory.com/programacion-extrema-que-es-y-principios-basicos>
- Desarrollo Ágil Kanban). Recuperado de <http://www.desarrolloweb.com/articulos/desarrollo-agil-kanban.html>

11. ANEXOS

ANEXO A. ENCUESTA REALIZADA PARA LOS REQUERIMIENTOS DE LA APLICACIÓN.

ENCUESTA AGENDA MEDICA

Descripción del formulario

¿Posee usted o alguno de sus familiares un dispositivo inteligente?*

- Si
- No

¿Con que sistema operativo cuenta su teléfono inteligente o tableta que tiene para su uso personal?

Si tiene mas de un teléfono o tableta puede selecciona mas de una respuesta

- Android (Samsung, HTC, etc)
- iOS(iPhone, iPad, iPod)
- Blackberry
- Windows Phone (Nokia, etc)
- JAVA(Celulares baja gama)
- Symbian (Nokia)
- Otro:

¿Estaría interesado en una aplicación movil que le facilite la vida en cuanto a su salud?*

- Si
- No

¿Debe estar pendiente de la salud de más familiares aparte de la suya?*

- Si
- No

¿Pierde constantemente citas médicas porque olvida su fecha y hora?*

- Si
- No

¿Ha llegado a una cita médica y no recordó en que consultorio es o con que médico le tocó?*

- Si
- No

¿Le ha tocado devolverse hasta su hogar para recoger exámenes olvidados u otros documentos para su cita médica?*

- Si
- No

¿Le es interesante que en una aplicación pueda evaluar que tan bien le fue con su doctor para después sacar una cita médica nuevamente con él?*

- Si
- No

¿Se toma puntualmente sus medicamentos?*

- Si
- No

¿Tiene problemas para recordar tomar sus medicamentos?*

- Si
- No



¿Que le gustaría que la aplicación le recordara a la hora de tomar sus medicamentos aparte del nombre y la hora?*

Puede seleccionar mas de una respuesta

- Foto del medicamento
- Contraindicaciones
- Efectos secundarios
- Fechas de vencimiento
- Alergias a algunos medicamentos
- Otro:

¿Le gustaría que la aplicación le recordara al momento en el que se le están agotando los medicamentos?*

- Si
- No

¿Le gustaría que la aplicación le recordara cuando validar las fórmulas de los medicamentos?*

A la hora de terminados los medicamentos y sacar una nueva formula

- Si
- No

¿Le gustaría que la aplicación le recordara cuando es la fecha y el lugar de entrega de medicamentos?*

- Si
- No

¿Olvida frecuentemente las especificaciones que necesita a la hora de tomarse un examen?*

- Si
- No



¿Que le gustaría que la aplicación le recordara a la hora de tomarse los exámenes aparte de fecha, hora y lugar?^{*}
Puede seleccionar mas de una respuesta

- Fecha y horario para recoger el examen
- Especificaciones de la preparación para el examen (que llevar, que hacer el día anterior, etc.)
- Especificación de para que es cada examen
- Otro:

Correo electronico*

Para enviarle información de esta futura aplicación

Opción 1

Agregar elemento ▼

ANEXO B. CUESTIONARIO DE USABILIDAD DE LA APLICACIÓN AGENDA MÉDICA

Para cada elemento identificado a continuación, rodee con un círculo el número de la derecha que considere más acorde con su criterio de calidad del software Agenda Médica. Utilice la escala anterior para seleccionar el número de calidad.

Pregunta	Escala de importancia				
	Malo	Medio	NS/NC	Bueno	Excelente
Rendimiento	1	2	3	4	5
¿Su uso es simple?	1	2	3	4	5
Velocidad	1	2	3	4	5
¿La organización de los menús es lógica?	1	2	3	4	5
Accesibilidad	1	2	3	4	5
¿La base de medicamentos es útil?	1	2	3	4	5
La interfaz es fácil de usar	1	2	3	4	5
¿La ayuda dada por el sistema es útil?	1	2	3	4	5
¿El sistema de alertas es eficaz?	1	2	3	4	5
¿El sistema de consultas es entendible?	1	2	3	4	5
¿La base de exámenes es útil?	1	2	3	4	5
¿Envío de mensajes de texto es competente?	1	2	3	4	5
Los mensajes de error y alertas correspondientes son claros y entendibles.	1	2	3	4	5

¿Tiene alguna observación para el equipo de trabajo, así como mejoras para el software?

Sí ___ No___

Cual:

¿Pagaría por una versión Premium del software?

Sí ___ No___