

**UNIVERSIDAD CENTRAL DE VENEZUELA**  
**FACULTAD DE CIENCIAS**  
**ESCUELA DE COMPUTACIÓN**

**SISTEMA MÓVIL**  
**DE REFERENCIAS MÉDICAS. PROGRAMA SOS**  
**TELEMEDICINA PARA VENEZUELA**



Trabajo Especial de Grado  
presentado ante la Ilustre  
Universidad Central de Venezuela  
Por los Bachilleres:

**Vanessa Rodríguez Ossandón**

**Alberto Bay Ungría**

**Para optar por el título de**  
**Licenciado en Computación**

**Tutor: Ana Morales**

Caracas, mayo 2014

## DEDICATORIA

*A mi madre que ha sido mi gran apoyo y me enseñó que con constancia y perseverancia puedo alcanzar todas mis metas.*

*A mi padre, mi fiel compañero, quien me dio siempre su ejemplo de lucha y esfuerzo para convertirme en la mujer que soy.*

*A mi familia que ha sido la base de mi vida y mi mayor orgullo.*

*A todas las personas que puedan beneficiarse directa o indirectamente con este proyecto.*

*Vanessa*

*A mi madre Aracelis y mi padre Andres quienes han sido mi ejemplo a seguir y un gran apoyo durante toda mi vida, a ustedes les debo quien soy.*

*A mis hermanos Alejandro, Andrea y Arturo por motivarme y darme la mano en los momentos mas difíciles para que pueda cumplir esta meta.*

*A mi novia Alba quien ha estado siempre a mi lado y con paciencia y comprensión me ha apoyado incondicionalmente y a su familia que siempre ha estado para mí en los buenos y malos momentos.*

*A mis familiares que desde el cielo están acompañándome y dándome fuerzas para conseguir todos mis objetivos.*

*A mis amigos incondicionales que estuvieron siempre ahí para aconsejarme durante momentos difíciles, Oscar y Adrian.*

*A todos los compañeros con los que cursé materias y que de alguna u otra forma me apoyaron a lo largo de esta carrera especialmente a aquellos que aun puedo llamar amigos como Reinaldo, Jonathan y Vanessa.*

*Alberto*

## **AGRADECIMIENTOS**

A Dios.

A nuestras familias.

A Ana Morales.

A Robinson Rivas.

A Nancy Urbina.

A Héctor Arrechdera.

A Antonio Silva.

Y a todo el equipo del Programa SOS Telemedicina para Venezuela.

## **RESUMEN**

Este trabajo presenta la elaboración e implementación de un sistema móvil integrado al Sistema de Referencias Médicas del "Programa SOS Telemedicina para Venezuela". El sistema está basado en el desarrollo de dos aplicaciones: una aplicación Web desarrollada en el lenguaje HTML y una aplicación nativa del sistema operativo móvil Android, desarrollada en Java. Se adiciona un actor importante al sistema como es el "Médico Rural", quien se encuentra desde el ambulatorio y puede ahora formar parte de la comunicación y obtener diagnósticos específicos, ampliando el alcance del sistema existente. El sistema móvil resultante, puede adaptarse al módulo y permitir así su funcionamiento entre los diferentes centros asistenciales que forman parte del Programa, sin importar el lugar donde se encuentren, todo esto a través de los dispositivos móviles.

## **ABSTRACT**

This paper presents the development and implementation of a mobile system integrated to Medical Reference System of "Programa SOS Telemedicina para Venezuela". The system is based on the development of two applications: a web application developed in HTML and a native Android mobile operating system, developed in Java. A new actor is added to the system, the doctor who is in the rural ambulatories and can now be part of the communication and obtaining specific diagnoses, expanding the scope of the existing system. The resulting mobile system can be adapted to the module, which allows cooperation among the various medical centers that are part of the program, no matter where they are, all through mobile devices

## ÍNDICE TEMÁTICO

|  |           |
|--|-----------|
| <b>RESUMEN</b> .....   | <b>4</b>  |
| <b>ABSTRACT</b> .....  | <b>4</b>  |
| <b>ÍNDICE DE FIGURAS</b> .....   | <b>8</b>  |
| <b>ÍNDICE DE TABLAS</b> .....  | <b>11</b> |
| <b>INTRODUCCIÓN</b> .....  | <b>13</b> |
| <b>CAPÍTULO I – PROBLEMA DE INVESTIGACIÓN</b> .....  | <b>17</b> |
| 1.1. INTRODUCCIÓN.....   | 17        |
| 1.2. PLANTEAMIENTO DEL PROBLEMA.....   | 17        |
| 1.3. SOLUCIÓN PROPUESTA.....   | 19        |
| 1.4. OBJETIVOS.....  | 20        |
| 1.4.1. Objetivo General.....   | 20        |
| 1.4.2. Objetivos Específicos.....  | 20        |
| 1.5. JUSTIFICACIÓN.....  | 21        |
| 1.6. ALCANCES.....   | 21        |
| <b>CAPÍTULO II – MARCO CONCEPTUAL</b> .....  | <b>22</b> |
| 2.1. INTRODUCCIÓN.....   | 22        |
| 2.2. DISPOSITIVOS MÓVILES.....   | 22        |
| 2.2.1. Categorías de los Dispositivos móviles.....   | 23        |
| 2.2.2. Tipos de Dispositivos Móviles.....  | 24        |
| 2.2.3. Tipos de Aplicaciones para dispositivos móviles.....  | 28        |
| 2.2.3.1. SMS (Servicio de Mensajes Cortos, Short Message Service).....   | 29        |
| 2.2.3.2. Sitios Web para móviles.....  | 30        |
| 2.2.3.3. Widgets móviles Web.....  | 31        |
| 2.2.3.4. Aplicaciones Web móviles.....   | 31        |
| 2.2.3.5. Aplicaciones Nativas.....   | 32        |
| 2.2.4. Sistema Operativo Móvil Android.....  | 33        |
| 2.3. USO DE TELÉFONOS INTELIGENTES (SMARTPHONES) EN LA MEDICINA.....   | 41        |
| 2.4. ESTUDIO DE FACTIBILIDAD TÉCNICA PARA LA IMPLEMENTACIÓN DE UNA APLICACIÓN PARA DISPOSITIVOS MÓVILES.....   | 43        |
| 2.4.1. Análisis Funcional y Comparativo.....   | 44        |
| 2.4.2. Tipos de Aplicaciones para Dispositivos Móviles Analizados.....   | 44        |
| 2.4.3. Análisis Comparativo.....   | 45        |
| 2.5. FACTIBILIDAD TÉCNICA PARA LA IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL NATIVA PARA DISPOSITIVOS MÓVILES..... | 48        |
| 2.5.1. Análisis Funcional y Comparativo.....   | 48        |
| 2.5.2. Sistemas Operativos Analizados.....   | 49        |
| 2.5.3. Análisis Comparativo desde el punto de vista del usuario.....   | 50        |

|  |    |
|--|----|
| 2.5.4. Análisis Comparativo desde el punto de vista del desarrollador .....  | 55 |
| 2.5.5. Análisis Comparativo desde el punto de vista de la penetración del mercado.....   | 60 |
| 2.6. TELEMEDICINA .....  | 60 |
| 2.6.1. Servicios de la Telemedicina .....  | 62 |
| 2.6.1.1. Asistencia remota:.....   | 62 |
| 2.7. SOS TELEMEDICINA PARA VENEZUELA .....   | 63 |
| 2.7.1. Triage .....  | 63 |
| 2.7.2. Referencia Médica .....   | 64 |
| 2.8. ANTECEDENTES .....  | 64 |
| 2.8.1. “Sistema de Historia Médica Electrónica para SOS Telemedicina”.....   | 65 |
| 2.8.2. “Desarrollo de un Sistema Web de Telemedicina para proveer Segunda Opinión Médica Especializada” .....                                    | 66 |
| 2.8.3. “Apoyo Tecnológico en el diseño, desarrollo, instalación y configuración de aplicaciones, Programa SOS Telemedicina para Venezuela”,..... | 67 |

### **CAPÍTULO III – MARCO METODOLÓGICO ..... 68**

|                                      |    |
|--------------------------------------|----|
| 3.1. INTRODUCCIÓN .....              | 68 |
| 3.2. METODOLOGÍA DE DESARROLLO ..... | 68 |
| 3.3. ETAPAS DE DESARROLLO .....      | 70 |
| 3.3.1. Etapa I. Planificación .....  | 70 |
| 3.3.2. Etapa II. Diseño.....         | 71 |
| 3.3.3. Etapa III. Codificación ..... | 71 |
| 3.3.4. Etapa IV. Pruebas .....       | 73 |

### **CAPÍTULO IV – MARCO APLICATIVO..... 74**

|   |     |
|---|-----|
| 4.1. INTRODUCCIÓN .....                                     | 74  |
| 4.2. CONTEXTO DEL PROYECTO.....                             | 74  |
| 4.3. DETALLE DE LA SOLUCIÓN.....                            | 74  |
| 4.4. ETAPAS DE DESARROLLO APLICACIÓN WEB MÓVIL.....         | 80  |
| 4.4.1. Etapa I. Planificación .....                         | 80  |
| 4.4.2. Etapa II. Diseño.....                                | 80  |
| 4.4.3. Etapa III. Codificación .....                        | 93  |
| 4.4.3.1. Arquitectura.....                                  | 94  |
| 4.4.3.2. HTML5 (hypertext Markup Language, versión 5).....  | 94  |
| 4.4.3.3. Javascript.....                                    | 95  |
| 4.4.3.4. AJAX (Asynchronous javascript And XML ) .....      | 96  |
| 4.4.3.5. JQuery.....  | 96  |
| 4.4.3.6. JQuery Mobile .....                                | 96  |
| 4.4.3.7. PHP (Hypertext Pre-processor).....                 | 97  |
| 4.4.3.8. Sistema Manejador de Base de Datos mysql.....      | 97  |
| 4.4.3.9. Wampserver.....                                    | 97  |
| 4.4.3.10. Módulos y funcionalidades.....                    | 100 |
| 4.4.4. Etapa IV. Pruebas .....                              | 117 |
| 4.5. ETAPAS DE DESARROLLO APLICACIÓN NATIVA EN ANDROID..... | 125 |
| 4.5.1. Etapa I. Planificación.....                          | 125 |

|  |  |            |
|--|--|------------|
| 4.5.2.   | Etapa II. Diseño .....   | 125        |
| 4.5.3.   | Etapa III. Codificación.....                                       | 130        |
| 4.5.3.1.   | Arquitectura .....   | 130        |
| 4.5.3.2.   | Sistema Operativo Android .....                                    | 131        |
| 4.5.3.3.   | Java.....  | 132        |
| 4.5.3.4.   | Entorno de desarrollo Android SDK (Software Development Kit) ..... | 132        |
| 4.5.3.5.   | XML (extensible Markup Language) .....                             | 133        |
| 4.5.3.6.   | JSON (Java Script Object Notation).....                            | 133        |
| 4.5.3.7.   | Sqlite .....   | 135        |
| 4.5.3.15.  | Servicios Web.....   | 135        |
| 4.5.3.16.  | Implementación de Servicio Web REST desde Android .....            | 137        |
| 4.5.3.17.  | Manejo de Base de datos sqlite .....                               | 141        |
| 4.5.3.18.  | Sincronización de la aplicación con el servidor .....              | 143        |
| 4.5.3.19.  | Parseo de Datos con JSON.....                                      | 145        |
| 4.5.3.20.  | Funcionalidades Aplicación Android .....                           | 146        |
| 4.5.4.   | Etapa IV. Pruebas .....  | 166        |
| <br><b>CAPÍTULO V – RESULTADOS GENERALES .....</b> |  | <b>172</b> |
| <br><b>CAPÍTULO VI – CONCLUSIONES.....</b>         |  | <b>176</b> |
| <br><b>REFERENCIAS BIBLIOGRÁFICAS.....</b>         |  | <b>180</b> |
| <br><b>APÉNDICE A – DOCUMENTACIÓN GRÁFICA.....</b> |  | <b>183</b> |

## ÍNDICE DE FIGURAS

|   |     |
|---|-----|
| Figura 1. Categorías de los Dispositivos Móviles.....   | 23  |
| Figura 2. Computadora Portátil ( <i>Laptop</i> ).....   | 24  |
| Figura 3. Teléfonos Móviles ( <i>Cell Phones</i> ).....                                       | 26  |
| Figura 4. Teléfonos Inteligentes ( <i>Smartphones</i> ).....                                  | 27  |
| Figura 5. Tabletas ( <i>Tablets</i> ).....  | 28  |
| Figura 6. Arquitectura de Android .....   | 34  |
| Figura 7. Directorio SRC Android .....  | 37  |
| Figura 8. Directorio RES Android .....  | 38  |
| Figura 9. Directorio BIN Android.....   | 39  |
| Figura 10. Directorio GEN Android.....  | 40  |
| Figura 11. Android Manifest.....  | 41  |
| Figura 12. Actividades Android.....   | 41  |
| Figura 13. Evolución de los diferentes Sistemas Operativos y su penetración en el mercado.... | 60  |
| Figura 14. Etapas De La Metodología XP .....  | 70  |
| Figura 15. Arquitectura del sistema móvil de Referencias Médicas.....                         | 72  |
| Figura 16. Flujo de interacción de MVC.....   | 73  |
| Figura 17. Diagrama de casos de uso nivel 0 .....   | 82  |
| Figura 18. Diagrama de casos de uso Nivel 1 .....   | 83  |
| Figura 19. Diagrama de clases del sistema móvil de Referencias Médicas.....                   | 87  |
| Figura 20. Diagrama de actividad para el caso de uso “Ingresar al Sistema” .....              | 88  |
| Figura 21. Máquina de Estados del Sistema de Referencias Médicas.....                         | 89  |
| Figura 22. Interfaz “Médico Triage” .....   | 91  |
| Figura 23. Menú Principal Médico Triage .....   | 92  |
| Figura 24. Menú Principal Médico Rural.....   | 93  |
| Figura 25. Arquitectura aplicación Web Móvil.....   | 94  |
| Figura 26. Configuración de parámetros.....   | 99  |
| Figura 27. Pantalla de Inicio.....  | 101 |

|  |     |
|--|-----|
| Figura 28. Detalle de Opinión .....                                | 102 |
| Figura 29. Pantalla Casos Nuevos.....                              | 105 |
| Figura 30. Pantalla Asignar / Reasignar Caso.....                  | 107 |
| Figura 31. Pantalla Casos Nuevos.....                              | 111 |
| Figura 32. Nuevo Registro “Crear Caso”.....                        | 115 |
| Figura 33. Pantalla Cerrar Casos.....                              | 117 |
| Figura 34. Diagrama de casos de uso nivel 0. ....                  | 126 |
| Figura 35. Diagrama de casos de uso nivel 1. ....                  | 127 |
| Figura 36. Arquitectura Sistema Móvil Nativo “Android” .....       | 131 |
| Figura 37. Código de creación de objeto JSON.....                  | 134 |
| Figura 38. Función httpPostRequest ().....                         | 138 |
| Figura 39. Objeto JSON.....  | 139 |
| Figura 40. Función httpGetRequest ( ).....                         | 139 |
| Figura 41. Función HttpGet() .....                                 | 140 |
| Figura 42. Datos de Petición al servidor .....                     | 140 |
| Figura 43. Objeto JSON de retorno.....                             | 141 |
| Figura 44. Clase MySQLiteOpenHelper. SQLite .....                  | 142 |
| Figura 45. Funciones SQLite.....                                   | 142 |
| Figura 46. Función onPreExecute().....                             | 143 |
| Figura 47. Función doInBackground ().....                          | 144 |
| Figura 48. Función onPostExecute().....                            | 144 |
| Figura 49. Objeto Json.....  | 145 |
| Figura 50. JSONArray ().....                                       | 146 |
| Figura 51. Pantalla de Bienvenida (Splash Screen).....             | 147 |
| Figura 52. Actividad creada en el archivo AndoridManifest.xml..... | 147 |
| Figura 53. Interfaz de Ingreso al Sistema.....                     | 148 |
| Figura 54. Objeto Json. Validación de usuario .....                | 149 |
| Figura 55. Validación de Usuario .....                             | 150 |
| Figura 56. Variable SahredPreferences.....                         | 151 |

|   |     |
|---|-----|
| Figura 57. Interfaz de Usuario Medico Rural.....                          | 151 |
| Figura 58. Descripción de Pantalla Crear Caso.....                        | 152 |
| Figura 59. Código para cargar imagen al servidor Android.....             | 153 |
| Figura 60. Seleccionador de Imagen de La Galería .....                    | 153 |
| Figura 61. Código para capturar imagen desde el dispositivo. Android..... | 154 |
| Figura 62. Pantalla Vista Previa de Imagen.....                           | 154 |
| Figura 63. Pantalla Casos Por Enviar .....                                | 156 |
| Figura 64. Pantalla de Casos Abiertos.....                                | 158 |
| Figura 65. Pantalla Detalle de Casos.....                                 | 159 |
| Figura 66. Pantalla Sincronizar Datos Con El Servidor SOS.....            | 161 |
| Figura 67. Pantalla Cerrar Sesión.....                                    | 165 |
| Figura 68. Arquitectura General de integración entre Sistemas .....       | 174 |

## ÍNDICE DE TABLAS

|   |     |
|---|-----|
| Tabla 1. Características técnicas que tienen los sistemas operativos móviles..... | 59  |
| tabla 2. Aspectos funcionales de los sistemas operativos .....                    | 78  |
| tabla 3. Actores identificados .....  | 81  |
| tabla 4. Casos de uso nivel 0 .....   | 82  |
| tabla 5. Casos de uso nivel 1 .....   | 83  |
| tabla 5. Casos de uso nivel 1 (continuación) .....                                | 84  |
| tabla 5.1. Casos de uso nivel 1 (continuación) .....                              | 85  |
| tabla 5.2. Casos de uso nivel 1 (continuación) .....                              | 86  |
| tabla 6. Prueba de funcionamiento 1 .....   | 118 |
| tabla 7. Prueba de funcionamiento 2 .....   | 118 |
| tabla 8. Prueba de funcionamiento 3 .....   | 119 |
| tabla 9. Prueba de funcionamiento 4 .....   | 119 |
| tabla 10. Prueba de funcionamiento 5.....   | 120 |
| tabla 11. Prueba de funcionamiento 6.....   | 120 |
| tabla 12. Prueba de funcionamiento 7.....   | 121 |
| tabla 13. Prueba de funcionamiento 8.....   | 121 |
| tabla 14. Prueba de funcionamiento 9.....   | 122 |
| tabla 15. Prueba de funcionamiento 10.....  | 122 |
| tabla 16. Prueba de funcionamiento 11.....  | 123 |
| tabla 17. Prueba de funcionamiento 12.....  | 123 |
| tabla 18. Prueba de funcionamiento 13.....  | 124 |
| tabla 19. Prueba de funcionamiento 14.....  | 124 |
| tabla 20. Casos de uso nivel 0. Android.....                                      | 127 |
| tabla 21. Casos de uso nivel 1. Android.....                                      | 128 |
| tabla 21.1. Casos de uso nivel 1. Android (continuación).....                     | 129 |
| tabla 22. Prueba de funcionamiento 1. Android.....                                | 166 |

|   |     |
|---|-----|
| tabla 23. Prueba de funcionamiento 2. Android.....  | 167 |
| tabla 24. Prueba de funcionamiento 3. Android.....  | 167 |
| tabla 25. Prueba de funcionamiento 4. Android.....  | 168 |
| tabla 26. Prueba de funcionamiento 5. Android.....  | 168 |
| tabla 27. Prueba de funcionamiento 6. Android.....  | 169 |
| tabla 28. Prueba de funcionamiento 7. Android.....  | 169 |
| tabla 29. Prueba de funcionamiento 8. Android.....  | 170 |
| tabla 30. Prueba de funcionamiento 9. Android.....  | 170 |
| tabla 31. Prueba de funcionamiento 10. Android..... | 171 |
| tabla 32. Prueba de funcionamiento 11. Android..... | 171 |

## INTRODUCCIÓN

En la actualidad resulta difícil caminar por las calles de las principales ciudades de Venezuela o del mundo, sin darse cuenta de la gran cantidad de personas que poseen algún tipo de dispositivo móvil, ya sea un teléfono celular (*Cell Phones*), una agenda electrónica (*PDA, Personal Digital Assitant*), una computadora portátil (*laptop*), una tableta (*tablet*) ó un teléfono inteligente (*Smartphone*). De igual forma sucede al caminar por los pasillos de clínicas, hospitales y centros de asistencia, donde se evidencia la gran cantidad de personal médico que posee alguno de estos dispositivos.

Los dispositivos móviles son pequeños computadores de mano, que permiten a sus usuarios realizar diferentes actividades mientras se desplazan de un lugar a otro. Por esta razón, este tipo de dispositivo tiene una gran cantidad de aplicaciones en el campo de la medicina moderna. Gracias al uso de dispositivos móviles los doctores pueden tener acceso a información sin importar el lugar donde se encuentren. Esta información incluye material de referencia clínico, libros de texto, material de investigación, prescripción de medicamentos, historial médico de los pacientes, entre otras.

Hoy en día las tecnologías de información y comunicación, así como el uso de dispositivos móviles se han incorporando a todas las ramas del saber, en especial el campo de la medicina ha resultado muy favorecido. Se ha podido evidenciar la variedad de usos que los profesionales de la industria médica pueden darle a los dispositivos móviles, así como los beneficios que aportan las aplicaciones médicas tanto para los profesionales de la salud como para los pacientes.

SOS Telemedicina para Venezuela [1] es un programa que contempla el diseño, desarrollo e implementación de las soluciones tecnológicas que permitan acercar oportunamente servicios especializados de salud de calidad y sin salir de su región a la población, en particular en zonas remotas del país, tiene como objetivo, entre otros, establecer una red de telemedicina a nivel nacional.

El programa SOS Telemedicina para Venezuela cuenta con el Ssistema de Referencias Médicas [2] en su versión web, el cual posee como objetivo principal, gestionar las referencias de la historia médica integral de un paciente, a través de un módulo Web que posibilita el apoyo para diagnosticar a un paciente mediante la solicitud de una referencia médica a especialistas que se encuentran en un lugar distante al del centro asistencial.

Con el fin de ampliar y fortalecer el programa SOS Telemedicina para Venezuela se planteó el diseño, desarrollo e implementación de un sistema móvil, que al integrarse al Sistema de Referencias Médicas, permita que los médicos y especialistas puedan realizar comentarios y segundas opiniones acerca del estado de salud de un paciente y la patología que padece, sin importar el lugar en el que se encuentren, haciendo uso de la tecnología móvil.

El sistema móvil representa una herramienta de gran utilidad para el Programa SOS Telemedicina para Venezuela, ya que complementa el Sistema de Referencias Médicas incorporando a éste, el uso desde dispositivos móviles. Entre las ventajas del desarrollo del sistema móvil se encuentran:

- Aumenta el alcance y beneficios que ofrece el Programa SOS Telemedicina para Venezuela.
- Suministrar opciones móviles al Sistema de Referencias Médicas, para el personal de salud de los diferentes centros de adscritos al Programa SOS Telemedicina para Venezuela.
- Permite la comunicación web a través de dispositivos móviles entre médicos rurales y médicos especialistas adscritos al Programa SOS Telemedicina para Venezuela.
- Ofrece acceso los recursos del Sistema de Referencias Medicas a través de dispositivos móviles.
- Posibilidad de que el tiempo transcurrido entre la llegada de un caso, y la emisión de una segunda opinión disminuya con el uso de la tecnología móvil.
- Capacidad para solventar diferentes patologías en pacientes sin importar el lugar en donde se encuentre el médico especialista, sólo haciendo uso de su dispositivo móvil.

En este trabajo se evalúan los aspectos técnicos relevantes para la implementación de éste sistema, tomando en cuenta las soluciones existentes y los requerimientos específicos del Programa SOS Telemedicina para Venezuela. Se presenta además la metodología y arquitecturas empleadas para la integración de cada una de las herramientas que se adaptan adecuadamente al contexto de la Organización. Las herramientas seleccionadas para la integración surgen de un análisis de factibilidad mostrado en el capítulo IV, donde se evaluaron tanto las técnicas para la implementación de una aplicación web móvil y aplicación móvil nativa.

Para éste desarrollo se implementaron dos soluciones:

La primera solución es el desarrollo de una versión web móvil del sistema, desde donde se podrá ingresar a la aplicación sin restricciones referente a cual sistema operativo se esté utilizando, ya que el ingreso se hará desde un navegador web móvil.

La segunda solución es la aplicación en el sistema operativo Android, la cual está implementada sólo para el usuario Médico Rural debido a que ofrece principalmente ventajas a este usuario permitiéndole uso de la aplicación sin conexión a Internet (principal problema de los centros asistenciales) además este sistema operativo por ser de tecnología abierta y adaptable permite realizar cualquier cambio que se requiera hacer durante el desarrollo, lo cual garantiza que se realicen solamente las actividades que son necesarias y útiles para el proyecto que se está trabajando, para de esta manera lograr la culminación exitosa del mismo.

Este trabajo especial de grado está conformado por los siguientes capítulos:

- **Capítulo I. Planteamiento del Problema**

En este capítulo se plantea la problemática actual y las limitaciones con respecto a la comunicación con los ambulatorios rurales y el personal médico especializado, además se definen los objetivos, el alcance y la solución propuesta de esta investigación.

- **Capítulo II. Marco Conceptual**

En este capítulo se definen todos los conceptos relacionados con los dispositivos móviles, telemedicina y aplicaciones en el ámbito de salud, además se describe el Programa SOS Telemedicina para Venezuela y los sistemas que lo conforman.

- **Capítulo III. Marco Metodológico**

En este capítulo se describe la metodología utilizada y las diferentes etapas en las que se llevo a cabo el desarrollo del sistema.

- **Capítulo IV. Marco Aplicativo**

En este capítulo se presentan el detalle de las diferentes etapas de desarrollo por las que se llevaron a cabo las aplicaciones desarrolladas, además se describen las pruebas realizadas y los resultados de las mismas.

- **Capítulo V. Resultados Generales**

En este capítulo se presentan los resultados generales obtenidos con el desarrollo de ésta investigación.

- **Capítulo VI. Conclusiones**

En este capítulo se detallan las conclusiones generales de esta investigación, las limitaciones existentes y las recomendaciones para trabajos futuros.

# CAPÍTULO I – PROBLEMA DE INVESTIGACIÓN

## 1.1. Introducción

En este capítulo se plantea el problema en relación con las limitaciones de la comunicación entre médicos rurales que se encuentran en los centros asistenciales y los médicos especialistas adscritos al Programa SOS Telemedicina para Venezuela para acortar los tiempos de respuesta y emitir de manera efectiva las diferentes patologías que presentan los pacientes. Seguidamente se expone la solución propuesta, se señala el objetivo general de éste trabajo y los objetivos específicos, además se justifica la realización de éste trabajo y se delimita su alcance.

## 1.2. Planteamiento del problema

Los ambulatorios de atención primaria son los encargados de prestar atención médica inmediata a los pacientes sin necesidad que éstos deban ser internados, en Venezuela cuentan con un médico permanente, un conjunto de residentes, estudiantes del último año de la carrera de medicina que se encuentran realizando la “Residencia” o personal de salud: enfermeras. Todos en el ejercicio de su trabajo pueden necesitar el apoyo de especialistas para lograr corroborar una impresión diagnóstica, para recibir indicaciones en cuanto al tratamiento adecuado para restablecer la salud de un paciente de su comunidad.

Actualmente, estas segundas opiniones médicas se obtienen mediante el contacto directo entre el paciente y el especialista, vía telefónica, a través del correo electrónico ó mediante aplicaciones automatizadas, como por ejemplo: el Sistema de Referencias Médicas del programa SOS Telemedicina para Venezuela (SOS-TRIAJE) [2], pudiendo con los dos últimos intercambiar imágenes digitales, resultados de laboratorio y otra información médica acerca del paciente.

Obtener una segunda opinión médica de un especialista ubicado a unos cuantos cientos de kilómetros no es siempre una tarea sencilla. Por lo que es importante contar con múltiples opciones para lograr el objetivo y mejorar las condiciones en el sistema de salud actual, por su parte contactar al especialista vía telefónica permite obtener de manera más rápida el estado del paciente, para así poder obtener una respuesta inmediata y el diagnóstico por parte del especialista. Al utilizar el correo electrónico, se podría realizar un análisis extensivo de la situación actual del paciente y a su vez permitiría adjuntar imágenes a la historia del paciente

para que así el especialista tenga mayor conocimiento y pueda realizar diagnósticos más precisos.

Obtener una referencia médica acerca del estado de un paciente significa que un especialista pueda emitir su diagnóstico y especificar el tratamiento indicado que requiere el paciente según la patología que presenta.

El Programa SOS Telemedicina para Venezuela [1], es un proyecto que contempla el diseño, desarrollo e implementación de las soluciones tecnológicas que permitan acercar oportunamente servicios especializados de salud de calidad y sin salir de su región a la población, en particular en zonas remotas del país.

El Sistema de Referencias Médicas del Programa SOS Telemedicina para Venezuela (SOS-TRIAJE) [2], permite realizar el intercambio de información sobre el estado de salud de un paciente entre los ambulatorios y centros asistenciales adscritos al Programa, donde se requiere de la opinión de un especialista que refleje su impresión diagnóstica sobre el caso, de ésta forma posibilita la pronta atención y la toma de decisiones más certeras sobre el cuadro clínico que se presente. Se supera así la limitante relacionada con la falta de atención y precisión en el diagnóstico sobre la salud de un paciente que se encuentra en zonas rurales y remotas.

Actualmente el Sistema SOS-TRIAJE no cuenta con un sistema móvil, que haga uso de diferentes lenguajes y herramientas que permitan la transformación e intercambio de los datos de manera más rápida y eficiente, lo cual no sólo facilitaría el proceso de segunda opinión permitiendo aumentar los beneficios y usos del Programa SOS Telemedicina para Venezuela si no que a su vez representa un beneficio real para los centros asistenciales del país.

Si se analiza la utilización del Sistema SOS-TRIAJE bajo un ambiente móvil, se permite a los médicos rurales una interacción con aquellos especialistas que con sólo ingresar al sistema pueden conocer la historia del paciente, analizar imágenes, y documentos que permitan obtener esa segunda opinión médica y tomar las acciones correctivas necesarias.

La gran mayoría de los médicos y personal de salud que labora y forma parte del Programa SOS Telemedicina para Venezuela, posee un dispositivo móvil y resultaría de mayor utilidad poder consultar, modificar y analizar la situación que presenta un paciente desde sus dispositivos sin la necesidad de encontrarse frente a un computador en un determinado momento, ello permitiría proporcionar diagnósticos con mayor rapidez y así atender a una mayor cantidad de pacientes, de manera más eficiente y precisa.

Existe entonces la necesidad de la implantación de un sistema móvil que permita el intercambio de información entre médicos rurales que se encuentran en los centros asistenciales y los médicos especialistas adscritos al Programa SOS Telemedicina para Venezuela.

Una solución móvil Web puede beneficiar a todos los usuarios del Sistema SOS-TRIAJE que utilicen dispositivos móviles, y en particular para los Médicos Rurales de atención primaria que posean dispositivos con el sistema operativo móvil Android. Una aplicación que les permita superar las limitaciones con la conexión a Internet, y facilite así, la comunicación con los diferentes especialistas de la salud.

### **1.3. Solución propuesta**

Se propone del diseño, desarrollo e implementación de un sistema móvil, que se integre al Sistema de Referencias Médicas (SOS-TRIAJE) [2], y permita la comunicación eficiente entre los diferentes actores que intervienen en la atención oportuna de un paciente, y ofrecer así diferentes diagnósticos que permitan soluciones dentro de los diferentes centros de salud adscritos al Programa SOS Telemedicina para Venezuela.

El sistema móvil se presenta en dos partes; una aplicación Web para móviles que permita a los usuarios el manejo del módulo de Referencias Médicas desde cualquier dispositivo móvil con conexión a Internet visualizado de forma correcta en sus navegadores. Por otra parte, se realiza el desarrollo de una aplicación móvil nativa desarrollada bajo el sistema operativo Android, ofreciendo así el acceso y conectividad mediante los dispositivos con dicho sistema operativo únicamente para médicos rurales de atención primaria.

Para poder implementar ambas aplicaciones móviles se realiza un levantamiento de información que permita identificar elementos clave en el contexto y dominio de las segunda opiniones.

La realización de un sistema móvil de referencias médicas en salud no es una tarea sencilla, ya que es necesario evaluar el sistema web existente y determinar así las adaptaciones y mejoras que se pueden aplicar aprovechando las ventajas que ofrece la tecnología móvil. Es por esto que se analiza la problemática descrita en el punto anterior con los siguientes criterios principales:

- Primera: evaluar el Sistema de Referencias Médicas (SOS-TRIAJE) existente y adoptar así sus funcionalidades en un sistema móvil.
- Segunda: ésta se deriva de la primera, en vez de adoptar íntegramente el sistema existente e implantarlo de manera móvil, se realizarían modificaciones a nivel de

interfaces, de funcionalidades y de flujo de información solamente, pero no a nivel de estructura, de manera que sea adaptado a las necesidades de la tecnología móvil y poder complementar así el módulo ya existente.

Para optar por alguno de estos criterios es necesario previamente conocer las limitaciones que presenta la tecnología móvil con respecto a la tecnología tradicional. Así mismo, se realiza una evaluación completa sobre el sistema actual para determinar así las funcionalidades más importantes y llevarlas de este modo a un ambiente móvil. Esta evaluación es un prerrequisito para el eventual desarrollo y/o implantación de un sistema móvil de referencias médicas.

La evaluación se realiza sobre productos de tecnología de carácter heterogéneo, es por esto que se tienen en cuenta criterios que permiten unificar formatos y ponderar de manera objetiva las características, el funcionamiento y la pertinencia para este proyecto.

Posterior a la evaluación se realiza el desarrollo y/o implementación del sistema móvil de referencias médicas. Esto responde a la necesidad de poder mantener la comunicación efectiva entre los médicos rurales y especialistas en la atención de diferentes patologías que se presentan en los centros asistenciales de nuestro país. Es necesario destacar lo alejados que se encuentran los médicos rurales de los especialistas, y que además están limitados en la tecnología de acceso a Internet como por ejemplo la red cableada.

## **1.4. Objetivos**

### **1.4.1. Objetivo General**

Diseñar, desarrollar e implementar un sistema móvil que se integre al Sistema de Referencias Médicas del programa SOS Telemedicina para Venezuela.

### **1.4.2. Objetivos Específicos**

Para alcanzar el objetivo general planteado se establecen los siguientes objetivos específicos:

- Analizar en detalle la información del proceso asistencial llevado a cabo en los centros de salud y el dominio clínico en el ámbito de referencias médicas.
- Evaluar tecnologías de hardware y software empleadas en los sistemas automatizados del Programa SOS Telemedicina para Venezuela (SOS-HME y SOS-TRIAJE).

- Evaluar el Sistema de Referencias Médicas (SOS-TRIAJE), su estructura y funcionamiento.
- Diseñar, desarrollar e implementar el módulo de Referencias Médicas en una aplicación móvil Web para dispositivos móviles.
- Diseñar, desarrollar e implementar el módulo de Referencias Médicas en una aplicación móvil nativa para el sistema operativo Android.
- Integrar las aplicaciones móviles con el sistema (SOS-TRIAJE) permitiendo la comunicación entre sistemas.
- Realizar pruebas a las aplicaciones.

### **1.5. Justificación**

El uso de un sistema móvil de referencias médicas se justifica por las siguientes razones:

- Ayuda a fortalecer la comunicación entre los diferentes centros asistenciales y médicos especialistas, complementando el sistema de Referencias Médicas mediante el uso de dispositivos móviles.
- Permite obtener de manera más rápida y eficiente segundas opiniones médicas sobre patologías presentadas por los pacientes.
- Asegura confidencialidad de la información clínica mediante controles de acceso.
- Permite compartir la información clínica entre distintos sistemas.
- La información clínica puede ser consultada en diferentes lugares de forma simultánea.
- Monitoreo del estado de salud de un paciente ingresado en algún ambulatorio o centro asistencial.

### **1.6. Alcances**

En este proyecto se desarrollará un sistema móvil de referencias médicas que esté en capacidad de intercambiar información con otros sistemas, contempla el diseño e implementación de dos aplicaciones: aplicación web móvil y aplicación nativa en Android, ambas se integran únicamente al Sistema de Referencias Médicas (SOS-TRIAJE) [2] y permiten la creación de nuevos casos mediante un nuevo usuario: Médico Rural.

El diseño de la solución debe incorporar estándares propios del ámbito de la informática médica. El intercambio de información estipulado incluirá contenidos multimedia, además de compartir datos de tipo texto.

## **CAPÍTULO II – MARCO CONCEPTUAL**

### **2.1. Introducción**

Este Trabajo de Grado se basa en el apoyo tecnológico para el desarrollo, configuración e instalación de un sistema móvil, por lo cual en este capítulo se abordará la conceptualización y definición de cada uno de los conceptos necesarios para el entendimiento del mismo y de cada uno de los términos involucrados en el desenvolvimiento del proyecto.

En este capítulo se abarca todo lo relacionado con los dispositivos móviles actuales en el mercado, los sistemas operativos y los tipos de aplicaciones a desarrollar para dichos dispositivos, además se explica en detalle los conceptos de telemedicina y del Programa SOS Telemedicina para Venezuela, los módulos que lo conforman, en particular, el módulo de Referencias Medicas, describiendo sus orígenes, sus principales funciones y los elementos que la caracterizan. Se detalla la perspectiva de propiedad y uso actual.

De la misma manera se presentan lo antecedentes que sirvieron como base de esta investigación y que traen consigo una serie de valores agregados al programa, detallando así la necesidad de contar con dicho sistema móvil.

### **2.2. Dispositivos móviles**

El término “móvil” hace referencia al traslado de las tareas de un sitio a otro, es decir, de llevarlas a cabo independiente del lugar en el cual se encuentre. [3]

Básicamente, los dispositivos móviles son aquellos que pueden ser transportados y que pueden ser utilizados durante su transporte. Estos dispositivos se pueden considerar como computadores personales con todo el significado de la palabra, ya que disponen de capacidad de procesamiento y de almacenamiento de datos. [3]

Gracias a los avances tecnológicos y especialmente al desarrollo de las tecnologías inalámbricas, prácticamente todos los dispositivos móviles pueden conectarse sin cables a una red brindándoles la característica de ser móviles e inalámbricos. [3]

### 2.2.1. Categorías de los Dispositivos móviles

Dado el variado número de niveles de funcionalidad asociadas con dispositivos móviles, se realizó una clasificación de los mismos, por ello en el 2005, T38 y DuPont Global Mobility Innovation Team propusieron los siguientes estándares para la definición de dispositivos móviles. [3]

- **Dispositivo Móvil de Datos Limitados (*Limited Data Mobile Device*):** teléfonos móviles clásicos. Se caracterizan por tener una pantalla pequeña de tipo texto. Ofrecen servicios de datos generalmente limitados a SMS y acceso WAP.
- **Dispositivo Móvil de Datos Básicos (*Basic Data Mobile Device*):** se caracterizan por tener una pantalla de mediano tamaño, menú o navegación basada en iconos, y ofrecer acceso a email, lista de direcciones, SMS, y, en algunos casos, un navegador Web. Un típico ejemplo de este tipo de dispositivos son los teléfonos inteligentes (*Smartphones*).
- **Dispositivo Móvil de Datos Mejorados (*Enhanced Data Mobile Device*):** se caracterizan por tener pantallas de medianas a grandes (por encima de los 240 x 120 pixeles), navegación de tipo stylus, y que ofrecen las mismas características que el "Dispositivo Móvil de Datos Básicos" más aplicaciones nativas como aplicaciones de Microsoft Office Mobile (Word, Excel, PowerPoint) y aplicaciones corporativas usuales, en versión móvil, como Sap, portales intranet, etc.

En la figura 1 se aprecia la clasificación de los dispositivos móviles según sus categorías:

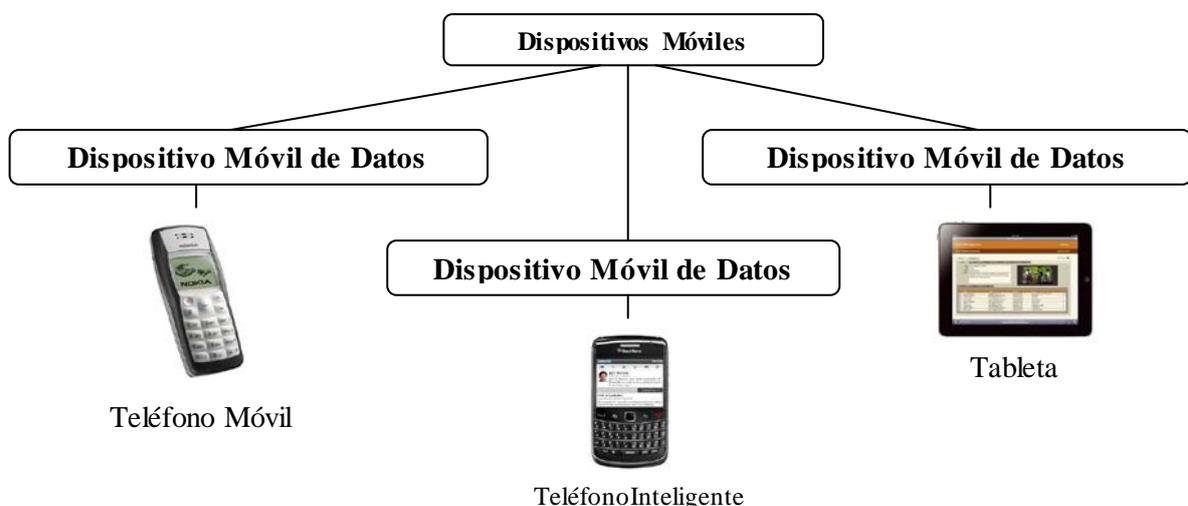


Figura 1. Categorías de los Dispositivos Móviles

### 2.2.2. Tipos de Dispositivos Móviles

En el siguiente apartado se presentan los diferentes tipos de dispositivos móviles presentes en la actualidad:

- **Computadoras Portátiles (*Laptops*)**

La computadora portátil es un dispositivo pequeño y ligero que puede llevar a cualquier lado, posee un teclado y una pantalla incorporados. Esto elimina los cables para conectar estos elementos. [3]

Las computadoras portátiles figuran entre los pocos productos de alta tecnología fabricados en grandes cantidades que permiten ofrecer al cliente la oportunidad de escoger las características que más requieran, son capaces de realizar la mayor parte o todas las tareas que realizan los computadores de escritorio, con similar capacidad; sumado también a que tienen la capacidad de operar por un período determinado sin estar conectadas a una red eléctrica. [3]

En la figura 2 se puede observar un ejemplo de una computadora portátil.



Figura 2. Computadora Portátil (*Laptop*)

Fuente: <http://www.hp.com>

- **Teléfonos móviles (*Cell Phones*)**

El teléfono móvil es un dispositivo inalámbrico electrónico para acceder y utilizar los servicios de la red de telefonía celular o móvil. Se denomina celular en la mayoría de países

latinoamericanos debido a que el servicio funciona mediante una red de celdas, donde cada antena repetidora de señal es una célula, si bien también existen redes telefónicas móviles satelitales. Su principal característica es su portabilidad, que permite comunicarse desde casi cualquier lugar. [4]

La principal función es la comunicación de voz, como el teléfono, su rápido desarrollo ha incorporado funciones adicionales como mensajería instantánea (sms), agenda, juegos, cámara fotográfica, acceso a Internet, reproducción de video e incluso GPS y reproductor mp3. [4]

La evolución del teléfono móvil ha permitido disminuir su tamaño y peso, desde el Motorola DynaTAC, el primer teléfono móvil en 1983 que pesaba 780 gramos, a los actuales más compactos y con mayores prestaciones de servicio. Además a lo largo de estos años se ha llevado a cabo el desarrollo de baterías más pequeñas y de mayor duración, pantallas más nítidas y de colores, la incorporación de software más amigable. Inicialmente los teléfonos móviles sólo permitían realizar llamadas de voz y enviar mensajes de texto. [4]

Conforme la tecnología fue avanzando se incluyeron nuevas aplicaciones como juegos, alarma, calculadora y acceso a Internet mediante páginas Web especialmente diseñadas para móviles. [4]

### **Características:**

- Son equipos compactos.
- Capacidad de manejo de llamadas y mensajes de texto.
- Algunos cuentan con cámaras digitales, generalmente de baja resolución.
- Muchos dispositivos de este tipo cuenta con soporte Bluetooth para manos libres.
- Generalmente cuentan con acceso a transmisiones de datos con tecnología WAP (*Wireless Application Protocol*, Protocolo de Aplicaciones Inalámbricas) para aplicaciones de video y música basados con las limitaciones inherentes del WAP.
- Explorador de Internet.
- Tarjeta de memoria para almacenar fotos, videos y música.
- Juegos, sonidos polifónicos y servicios básicos adicionales.

En la figura 3 se observa un ejemplo de varios teléfonos móviles.



Figura 3. Teléfonos Móviles (*Cell Phones*)

Fuente: <http://www.sudamerica-moviles.com>

- **Teléfonos inteligentes (*Smartphones*)**

Un teléfono inteligente es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de computación y conectividad que un teléfono móvil convencional. El término «inteligente» hace referencia a la capacidad de usarse como un computador de bolsillo, llegando incluso a remplazar a un computador personal en algunos casos. [5]

Casi todos los teléfonos inteligentes también permiten al usuario instalar programas adicionales, normalmente inclusive desde terceros hecho que dota a estos teléfonos de muchísimas aplicaciones en diferentes terrenos, pero algunos vendedores gustan de tildar a sus teléfonos como inteligentes aún cuando no tienen esa característica. [5]

Algunos ejemplos de teléfonos denominados inteligentes son: serie iPhone de Apple, Serie BlackBerry de Research In Motion, Serie Ascend de Huawei, Serie Defy de Motorola, Serie Optimus de LG, Serie Lumia de Nokia, Serie Galaxy Nexus de Google/Samsung, Serie One de HTC, Serie Xperia de Sony Mobile Communications, Serie Galaxy de Samsung. [5]

Entre otras características comunes está la función multitarea, el acceso a Internet vía Wi-fi o 3G, a los programas de agenda, a una cámara digital integrada, administración de contactos, acelerómetros, GPS y algunos programas de navegación así como ocasionalmente la

habilidad de leer documentos de negocios en variedad de formatos como PDF y Microsoft Office. [5]

En la figura 4 se puede apreciar un ejemplo de varios teléfonos inteligentes o *Smartphones*.



Figura 4. Teléfonos Inteligentes (*Smartphones*)

Fuente: <http://www.pcworld.com.mx>

- **Tableta (*Tablet*)**

Una tableta es un tipo de computadora portátil, de mayor tamaño que un teléfono inteligente o una PDA, integrado en una pantalla táctil (sencilla o multitáctil) con la que se interactúa primariamente con los dedos o una *pluma stylus* (pasiva o activa), sin necesidad de teclado físico ni ratón. Estos últimos se ven reemplazados por un teclado virtual y, en determinados modelos, por una mini-trackball integrada en uno de los bordes de la pantalla. [4]

El término puede aplicarse a una variedad de formatos que difieren en la posición de la pantalla con respecto a un teclado. El formato estándar se llama pizarra (*Slate*) y carece de teclado integrado aunque puede conectarse a uno inalámbrico (por ej., Bluetooth) o mediante un cable USB (muchos sistemas operativos reconocen directamente teclados y ratones USB). Otro formato es el portátil convertible, que dispone de un teclado físico que gira sobre una bisagra o se desliza debajo de la pantalla. Un tercer formato, denominado híbrido (como el HP COMPAQ TC1100), dispone de un teclado físico, pero puede separarse de él para comportarse como una pizarra. Por último los *Booklets* incluyen dos pantallas, al menos una de ellas táctil, mostrando en ella un teclado virtual. [4]

En la figura 5 se observa un ejemplo de algunas tabletas o *tablets*.



Figura 5. Tablet (Tablets)

Fuente: <http://www.pcworld.pe>

### 2.2.3. Tipos de Aplicaciones para dispositivos móviles

El desarrollo de aplicaciones móviles difiere del desarrollo de software tradicional en muchos aspectos, lo que provoca que las metodologías usadas para estos entornos también difieran de las del software clásico. Esto es porque el software móvil tiene que satisfacer una serie de requerimientos y condiciones que lo hace más complejo [6]:

- **Conectividad:** consideraciones tales como la disponibilidad, las desconexiones, la variabilidad del ancho de banda, la heterogeneidad de redes o los riesgos de seguridad deben tenerse especialmente en cuenta en este entorno de comunicaciones móviles.
- **Movilidad:** aquí influyen consideraciones como la migración de direcciones, alta latencia debido a cambio de estación base o la gestión de la información dependiente de localización. Sobre esta última, de hecho, se pueden implementar un sinnúmero de aplicaciones, pero la información de contexto asociada resulta muchas veces incompleta y varía frecuentemente.
- **Portabilidad:** la característica portabilidad de los dispositivos implica una serie de limitaciones físicas directamente relacionadas con el factor de forma de los mismos, como el tamaño de las pantallas, o del teclado.

- **Fragmentación de la industria:** la existencia de una considerable variedad de estándares, protocolos y tecnologías de red diferentes añaden complejidad al escenario del desarrollo móvil.
- **Capacidades limitadas de los terminales:** se incluyen factores como la baja potencia de cálculo o gráfica, los riesgos en la integridad de datos, las interfaces de usuario poco funcionales en muchos aspectos, la baja capacidad de almacenamiento, la duración de las baterías o la dificultad para el uso de periféricos en movilidad. Factores todos que, por otro lado, están evolucionando en la dirección de la convergencia de los ultra portátiles (*netbooks*) con los dispositivos inteligentes constituyendo cada vez menos un elemento diferencial.
- **Usabilidad:** las necesidades específicas de amplios y variados grupos de usuarios, combinados con la diversidad de plataformas tecnológicas y dispositivos, hacen que el diseño para todos se convierta en un requisito que genera una complejidad creciente difícil de acotar.

El diseño de sistemas de software móvil es, por tanto, es bastante más complejo que el tradicional, forzando a los investigadores a reconsiderar el uso de las metodologías actuales de desarrollo de software. [6]

La disponibilidad de las herramientas y la facilidad para construir y comercializar software para el móvil ha hecho que su número aumente considerablemente en la actualidad.

#### **2.2.3.1. SMS (Servicio de Mensajes Cortos, *Short Message Service*)**

La aplicación móvil más básica que se puede crear son las de tipo SMS. A pesar de que podría parecer extraño considerar las aplicaciones de mensajes de texto, son sin embargo un diseño con experiencia. Dada la ubicuidad de los dispositivos que soportan SMS, estas aplicaciones pueden ser herramientas útiles cuando se integra con otros tipos de aplicaciones móviles. [6]

Normalmente, el usuario envía una palabra clave a un código corto de pocos dígitos que resulta en la devolución de información o un enlace a contenidos de calidad.

Los usos más comunes de las aplicaciones SMS son la descarga de contenido móvil, tal como tonos de llamada e imágenes o para interactuar con bienes y servicios reales. Los mensajes SMS también pueden utilizarse para compra de tiempo en un estacionamiento o zona de pago.

Las ventajas de las aplicaciones SMS son:

- Trabajan en cualquier dispositivo móvil casi instantáneamente.
- Son útiles para el envío de alertas a tiempo al usuario.
- Se pueden incorporar en cualquier Web o de aplicación móvil.
- Pueden ser fáciles de configurar y administrar.

Las desventajas de las aplicaciones SMS son:

- Están limitados a 160 caracteres.
- Ofrecen una experiencia limitada basada en texto.
- Las implementaciones pueden resultar costosas.

#### **2.2.3.2. Sitios Web para móviles**

Es un sitio Web diseñado específicamente para dispositivos móviles, que no debe confundirse con la visualización de un sitio hecho para los navegadores en un móvil. Estos se caracterizan por la simple presentación de los enlaces de navegación que le llevará a una página de un nivel más profundo. [6]

Estos sitios Web móviles suelen tener un diseño simple y son típicamente de naturaleza informativa, que ofrece pocos de los elementos interactivos que se puede esperar de un sitio de Web para escritorio.

Motivado a que mejores navegadores móviles comenzaron a ser introducidos en plataformas de dispositivos como el iPhone y Android, la calidad de sitios Web para móviles comenzaron a mejorar de manera espectacular, y mejoró su utilización.

Las ventajas de sitios Web para móviles son:

- Son fáciles de crear, mantener y publicar.
- Se pueden utilizar todas las mismas herramientas y técnicas que en sitios de computadoras de escritorio.
- Casi todos los dispositivos móviles pueden ver sitios Web para móviles.

Las desventajas de sitios Web para móviles son:

- El soporte puede ser difícil a través de múltiples dispositivos.
- Ofrecen a los usuarios una experiencia limitada.
- Se pueden cargar las páginas lentamente, debido a la latencia de la red.

### **2.2.3.3. Widgets móviles Web**

Este es un componente de interfaz de usuario con un bloque independiente de código HTML que es ejecutado por el usuario de una manera particular. Básicamente, los Widgets Web para móviles son pequeñas aplicaciones Web que no se puede ejecutar por sí mismos. Puede crear atractivas experiencias de usuario que aprovechan las funciones del dispositivo y, en muchos casos, requieren ejecutarse mientras el dispositivo está en línea. [6]

Las ventajas de Widgets Web para móviles son:

- Son fáciles de crear, a partir de HTML, CSS, Java Script y el conocimiento.
- Pueden ser fácil de implementar a través de múltiples dispositivos.
- Ofrecen una experiencia de usuario mejorada y un diseño más rico, y algunos no requieren conexión.

Las desventajas de Widgets Web para móviles son:

- Por lo general requieren una plataforma de Widgets compatibles para ser instalado en el dispositivo.
- Generalmente no se puede ejecutar en cualquier navegador Web para móviles.
- En algunos casos, se requiere el aprendizaje adicional técnicas propias de Widgets.

### **2.2.3.4. Aplicaciones Web móviles**

Las aplicaciones Web móviles son aplicaciones móviles que no necesitan ser instaladas o compiladas en el dispositivo de destino. Usando XHTML, CSS y Java Script, son capaces de proporcionar una rica experiencia de la aplicación para el usuario final. Este tipo de aplicaciones permiten a los usuarios interactuar con el contenido en tiempo real, donde un clic o toque realiza una acción dentro de la vista. [6]

El desarrollo y uso de este tipo de aplicaciones explotó con la llegada del iPhone, y más tarde los dispositivos con Android. Esto es porque los fabricantes instalaron un movimiento para mejorar los navegadores de sus dispositivos, dotándolos de soporte estándar casi completo para XHTML, CSS, Java Script y Ajax para aprovechar este nuevo medio de aplicaciones.

Las ventajas de las aplicaciones Web para móviles son:

- Son fáciles de crear, a partir de HTML, CSS, Java Script y contenido.
- Son fáciles de implementar a través de múltiples dispositivos.
- Ofrecen una mejor experiencia de usuario, aprovechando las características del dispositivo.
- El contenido es accesible en cualquier navegador Web para móviles.

Las desventajas de las aplicaciones Web para móviles son:

- La experiencia óptima podría no estar disponible en todos los terminales.
- Puede resultar un reto (aunque no imposible) el apoyo a través de múltiples dispositivos.
- No siempre se cuenta con el apoyo nativo de aplicaciones, como el modo en línea, ubicación, operaciones de búsqueda, sistema de ficheros de acceso, cámaras, etc.

### **2.2.3.5. Aplicaciones Nativas**

El medio de aplicación más antigua y común. Estas aplicaciones nativas o de la plataforma se construyen específicamente para los dispositivos que ejecutan la plataforma de que se trate. El más común de todas las plataformas es Java ME. [7]

Con la llegada de los *Smartphones*, las SDK de cada plataforma consiguen una implementación mucho más específica. Una capa del sistema operativo y APIs añadidos permiten a los desarrolladores implementar más fácilmente las tareas complejas de la API en lugar de escribir métodos a partir de cero.

La creación de aplicaciones utilizando una plataforma de aplicaciones significa decidir los dispositivos destino, teniendo medios de ensayos y certificación, y un método para distribuir la aplicación a los usuarios. La gran mayoría de las aplicaciones nativas están certificadas, se venden y distribuyen a través de un portal del operador o una tienda de aplicaciones.

Con las aplicaciones nativas que se desarrollan sobre la capa superior de la plataforma, se puede aprovechar la mayoría de las características del dispositivo.

Las ventajas de las aplicaciones nativas incluyen:

- Ofrecen la mejor experiencia de usuario, aprovechan las funciones del dispositivo incluso sin conexión.
- Son relativamente fácil de desarrollar para una sola plataforma.
- El cobrar por las aplicaciones es más fácil.

Los contras de las aplicaciones nativas incluyen:

- No son fácilmente portadas a otras plataformas móviles.
- El desarrollo, prueba y soporte de múltiples plataformas de dispositivo es costoso.
- Requieren certificación y distribución de un tercero que no tiene control.
- Generalmente se necesita compartir los ingresos con una o más terceras partes.

#### **2.2.4. Sistema Operativo Móvil Android**

Android es un sistema operativo móvil basado en Linux, que está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio. [8]

El anuncio del sistema Android se realizó el 5 de noviembre de 2007 junto con la creación de la Open Handset Alliance, Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto.

Este sistema por lo general maneja aplicaciones descargadas desde Google Play, aunque pueden ser instaladas por otros medios, a su vez busca, un modelo estandarizado de programación que simplifique las labores de creación de aplicaciones móviles y normalice las herramientas en el campo de la telefonía móvil. Al igual que ocurriera con Symbian, lo que se busca es que los programadores sólo tengan que desarrollar sus creaciones una única vez y así ésta sea compatible con diferentes terminales. [8]

El corazón de Android es el kernel Linux, donde se encuentran los *drivers* necesarios para el acceso al *hardware*, en concreto, para la gestión de la pantalla, el teclado, la cámara, la red Wi-Fi, el audio y la memoria Flash, entre otros.

En la figura 6 se muestra la composición de la arquitectura de Android.

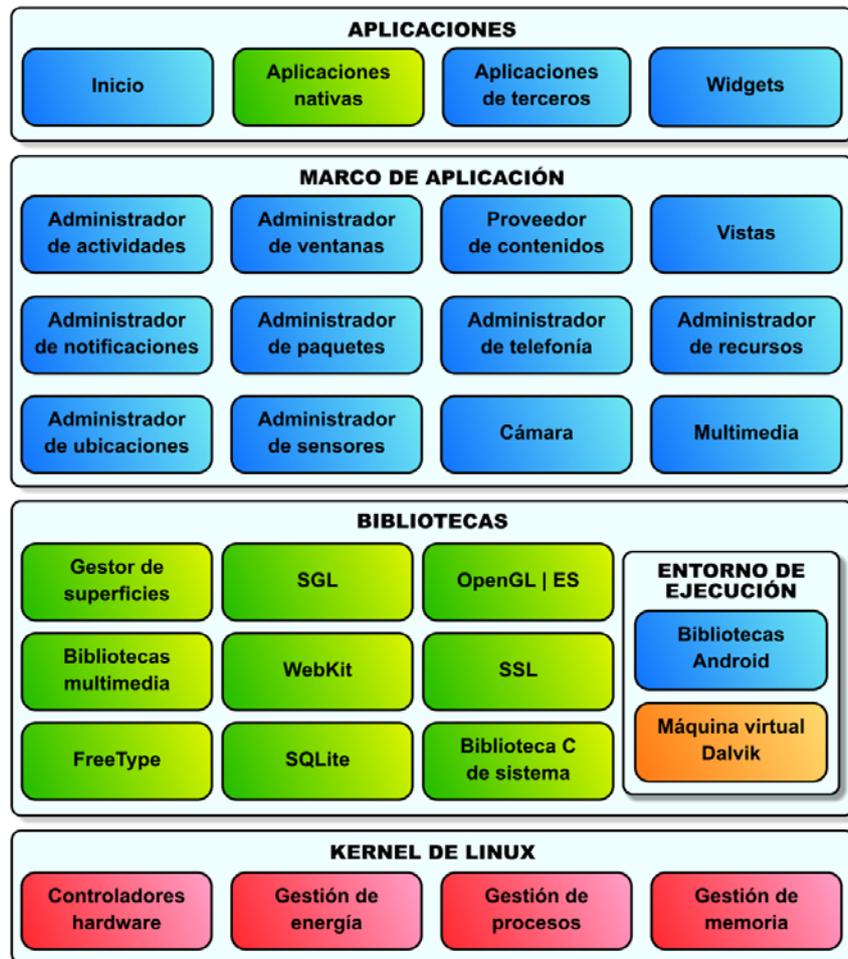


Figura 6. Arquitectura de Android

Fuente: <http://developer.android.com>

### Capas por la que está compuesta la arquitectura Android

#### Aplicaciones

Android se distribuye con un conjunto de aplicaciones básicas, como un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos, y otros. Las aplicaciones están escritas en el lenguaje de programación Java. [8]

## Framework de aplicaciones

Toda aplicación que es desarrollada en Android utiliza el mismo conjunto de API (Interfaz de Programación de Aplicaciones) y el mismo *framework* (marco de trabajo) que las utilizadas por las aplicaciones principales. [8] Entre las API más importantes se pueden encontrar las siguientes:

- **Activity Manager:** gestiona el ciclo de vida de las aplicaciones Android.
- **Window Manage:** gestiona las ventanas de las aplicaciones.
- **Telephone Manager:** incluye todas las API vinculadas a las funcionalidades propias del teléfono (llamadas, mensajes, etc.)
- **Content Providers:** permite a cualquier aplicación compartir sus datos con las demás aplicaciones de Android. Por ejemplo, gracias a esta API la información de contactos, agenda, mensajes, etc. será accesible para otras aplicaciones.
- **View System:** proporciona un gran número de elementos para poder construir interfaces de usuario (GUI), como listas, mosaicos, botones, check-boxes, tamaño de ventanas, control de las interfaces mediante tacto o teclado, etc. Incluye también algunas vistas estándar para las funcionalidades más frecuentes.
- **Location Manager:** posibilita a las aplicaciones la obtención de información de localización y posicionamiento.
- **Notification Manager:** mediante el cual las aplicaciones, usando un mismo formato, comunican al usuario eventos que ocurran durante su ejecución: una llamada entrante, un mensaje recibido, conexión Wi-Fi disponible, ubicación en un punto determinado, etc. Si llevan asociada alguna acción, en Android denominada Intent, (por ejemplo, atender una llamada recibida) ésta se activa mediante un simple clic.
- **XMPP Service:** colección de API para utilizar este protocolo de intercambio de mensajes basado en XML.

## Librerías

Android incluye un conjunto de librerías C/C++ utilizadas por los diversos componentes del sistema Android. Estas capacidades están expuestas a los desarrolladores a través del *framework* de aplicaciones. Entre las librerías más importantes en este nivel se pueden encontrar [8]:

- Librerías gráficas que incluyen OpenGL y SGL para manejo de gráficos 3D y 2D, respectivamente.
- SQLite motor de base de datos relacional. permite tener una base de datos local con la que gestionar todos los datos de la aplicación.
- Webkit, con el que puede integrar el motor de renderización Web que usan Safari, Chrome y el propio Android.
- Surface Manager provee gestión de las pantallas. Gestiona las ventanas pertenecientes a las distintas aplicaciones activas en cada momento
- SSL para mantener la seguridad sobre Internet.
- Media Framework para poder hacer uso de audio y vídeo, se debe usar esta librería.
- FreeType. Con ella se puede renderizar textos usando diferentes fuentes.
- La librería libc (standard C system library) incluye todas las cabeceras y funciones según el estándar del lenguaje C. Todas las demás librerías se definen en este lenguaje.

## Runtime Android

Cada aplicación Android se ejecuta en su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito para que un dispositivo pueda ejecutar varias máquinas virtuales de manera eficiente.

## Kernel de Linux

Android se basa en la versión Linux 2.6 para los servicios del núcleo del sistema, como la seguridad, gestión de memoria, gestión de procesos, pila de red, y el modelo del controlador. El kernel provee una abstracción entre el hardware y el resto de las capas. Entre los tipos de componentes más relevantes se pueden describir los siguientes [8]:

- **Activity:** es el encargado de gestionar la interfaz de usuario.
- **Service:** que se ejecutan totalmente en segundo plano.

- **Broadcast receiver:** componentes se ejecutan en segundo plano que reciben datos de cambios en el sistema.
- **Content provider:** necesario para la compartición de datos entre los distintos componentes de la aplicación.

### Estructura de una aplicación Android

Toda aplicación creada para Android sigue una misma estructura básica, que se compone del código fuente en sí, archivos de recursos y vistas, librerías de código y el *Android Manifest*.

- **Directorio SRC**

Es en donde se encuentran todas las clases programadas en JAVA. Es el directorio en donde se encuentra toda la lógica de la aplicación. Dentro de este directorio se pueden definir distintos paquetes y así dividir en capas las reglas de negocio del sistema. En la Figura 7 se puede apreciar en contenido del directorio SRC de Android.

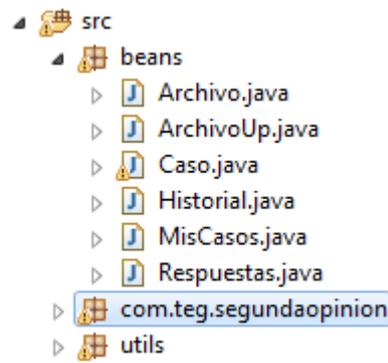


Figura 7. Directorio SRC Android

Fuente: <http://developer.android.com>

- **Android Library**

Aquí se encuentran todas las librerías propias del SDK de Android.

- **Directorios RES**

Se consideran recursos todos aquellos elementos que utiliza la aplicación que no forman parte del código fuente como lo son:

- Imágenes.
- Archivos de idiomas.
- Estilos.
- Ficheros XML.

Android permite definir fuera del código otros elementos como cadenas de texto, menús o incluso los diseños de las actividades. Esto se denomina externalización de recursos.

La externalización de recursos permite modificar dichos recursos de forma independiente permitiendo tener variantes del mismo recurso de forma que se pueda usar uno u otro según las características del dispositivo donde se ejecute la aplicación. Esta última característica es la más importante de todas, dado que permite que los textos de la aplicación se muestren en el idioma que tiene configurado el usuario o que la disposición de los elementos visuales se adapte al tamaño de la pantalla. En conclusión la externalización de recursos permite tener un desarrollo dinámico y una mejor implementación de los recursos de la aplicación en comparación a la forma tradicional en donde los recursos están codificados en archivos Java dentro del propio código fuente.

En la Figura 8 se puede apreciar un ejemplo de un directorio RES de una aplicación en Android.

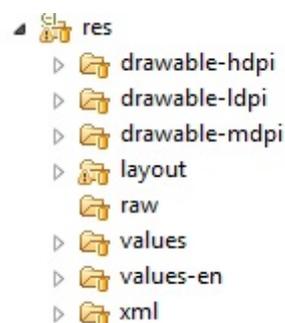


Figura 8. Directorio RES Android

Fuente: <http://developer.android.com>

- **Drawable:** directorio con todas las imágenes de la app. Se subdivide en múltiples carpetas que contienen las imágenes en distintas resoluciones y tamaños que se desplegarán en pantalla dependiendo del dispositivo usado.
- **Layout:** directorio en donde se encuentran las distintas interfaces visuales de la aplicación, es decir, los archivos con extensión `.xml` que contienen las pantallas asociadas a cada activity.
- **Values:** directorio con los xml del contenido de la aplicación. En ella pueden estar definidas las constantes de la aplicación, dando la posibilidad de tener una aplicación multi-idomas. Adicionalmente se pueden definir estilos para los componentes así como los valores para los campos de texto, títulos, botones entre otros

- **Directorio BIN**

Contiene los elementos compilados de la aplicación y otros ficheros auxiliares. Cabe destacar el fichero con extensión `“.apk”`, que es el ejecutable de la aplicación que se instalará en el dispositivo. En la Figura 9 se muestra el contenido del directorio bin.

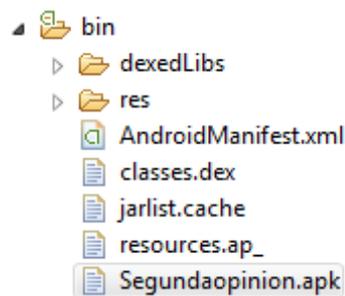


Figura 9. Directorio BIN Android

Fuente: <http://developer.android.com>

- **Directorio GEN**

Contiene una serie de elementos de código generados automáticamente al compilar el proyecto. Cada vez que se genera un proyecto, la maquinaria de compilación de Android genera una serie de ficheros fuente java dirigidos al control de los recursos de la aplicación. En la Figura 10 se aprecia el contenido del directorio Gen.

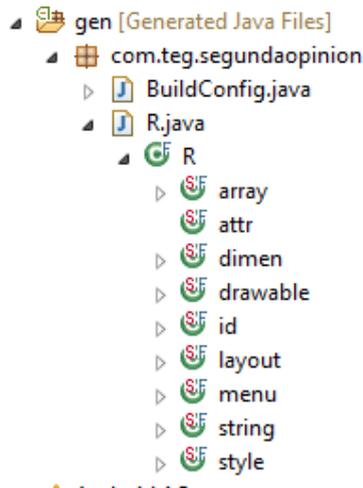


Figura 10. Directorio GEN Android

Fuente: <http://developer.android.com>

- **Directorio ASSESTS**

Contiene todos los demás ficheros auxiliares necesarios para la aplicación (y que se incluirán en su propio paquete), como por ejemplo ficheros de configuración, de datos, etc.

La diferencia entre los recursos incluidos en la carpeta `/res/raw/` y los incluidos en la carpeta `/assets/` es que para los primeros se generará un ID en la clase `R` y se deberá acceder a ellos con los diferentes métodos de acceso a recursos. Para los segundos sin embargo no se generarán ID y se podrá acceder a ellos por su ruta como a cualquier otro fichero del sistema. Se utiliza uno u otro según los requerimientos de la aplicación.

- **Directorio LIB**

Contendrá las librerías auxiliares, normalmente en formato `“.jar”` que se utilizarán en la aplicación Android.

- **Android Manifest**

Archivo básico de configuración de toda aplicación Android que está situado en la raíz de la aplicación. Contiene la definición en XML de los aspectos principales de la aplicación, como por ejemplo su identificación (nombre, versión, icono, ...), sus componentes (pantallas, mensajes, ...), las librerías auxiliares utilizadas, o los permisos necesarios para su ejecución.

En la Figura 11 se puede apreciar un ejemplo de la configuración en el AndroidManifest.xml.

```
<uses-permission /> //Permisos que el usuario debe aceptar al instalar
<permission /> //Permisos de la app
<uses-sdk /> //Configuración de las versiones soportadas.
<supports-screens /> //Pantallas soportadas
<compatible-screens /> //Pantallas compatibles
```

Figura 11. Android Manifest

Fuente: <http://developer.android.com>

Todas y cada una de las actividades de la aplicación deben ser declaradas en esta sección, sino no funcionarán. La actividad inicial de la aplicación debe ser declarada de tipo "LAUNCHER" como se puede apreciar en la Figura 12.

```
<application android:icon="@drawable/icon" android:label="@string/app_name">
//Declaración de todas las actividades de la app:
<activity android:name=".actividad1" android:label="@string/app_name">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

Figura 12. Actividades Android

Fuente: <http://developer.android.com>

### 2.3. Uso de Teléfonos Inteligentes (*Smartphones*) en la medicina

La tecnología móvil cada vez se vuelve más importante en todas las ramas del conocimiento y la medicina no podía quedar fuera de esta tendencia. En la actualidad, los smartphones cuentan con una serie de aplicaciones sumamente útiles para quienes estudian medicina y también para quienes son médicos. [9]

En el caso de la medicina, cada vez hay más aplicaciones médicas para dispositivos móviles, estas aplicaciones y periféricos son una realidad para los *smartphones*. Con el auge y la expansión del mercado de los *smartphones*, también ha avanzado enormemente el campo del desarrollo de aplicaciones para móviles cada vez más avanzadas. Según un estudio, el 72% de los médicos utilizan este tipo de teléfonos, y la cifra va en aumento. Además, el 95% de los médicos que usan *smartphones* utilizan aplicaciones médicas. [9]

Cada vez más se expanden los usos de los *smartphones*, es probable que dentro de poco todo médico tenga que contar con uno. Por otra parte, el desarrollo de recursos como las aplicaciones médicas pueden ser de gran utilidad, ya que permiten al paciente adquirir una cultura de prevención y responsabilidad frente a su salud, una actitud que, además de ser útil para él mismo, podría reducir de manera significativa el gasto sanitario. Además, se ha visto que cerca de la mitad de los tratamientos médicos no se terminan y muchos pacientes no siguen las recomendaciones de estilo de vida saludable que les aconseja el médico. [9]

Cabe recordar, no obstante, que las aplicaciones médicas son solo una herramienta complementaria y que en ningún caso deben sustituir el papel de los profesionales de la salud. Por este motivo, se recomienda usarlas como fuente de información y consulta, sin confiarse en exceso ni alarmarse

Algunas aplicaciones existentes para smartphones en el área de medicina son [9]:

- **LifeLens:** ésta aplicación es capaz de permitir a los médicos elaborar un diagnóstico de malaria con el uso de una simple fotografía de teléfono móvil. El prototipo usado en el experimento es un terminal Samsung Focus (un Samsung Omnia 7), que funciona bajo el sistema operativo Windows Phone 7, y que lleva añadidas unas lentes microscópicas a la propia cámara del teléfono.

Con este montaje el procedimiento a realizar es el siguiente: Se toma una fotografía de una muestra de sangre, se procesa la imagen para encontrar parásitos de malaria en ella, se dibujan cuadrados rojos alrededor de los grupos de malaria encontrados, se determinan cuantos grupos se han encontrado.

Ésta aplicación elimina la barrera que supone tener que trasladar al paciente a los centros de examen (sobre todo en sitios remotos o poco desarrollados) y permite que el teléfono móvil sea el transportado a la zona.

- **OnTrack:** es una aplicación que ayuda a los diabéticos a sobrellevar la diabetes. Con seguimiento actualizado sobre cuestiones de salud como glucosa en la sangre, alimentación, dieta balanceada, medicación, presión de la sangre y pulso. Además recomienda rutinas de ejercicio y lleva también un control de peso.

La aplicación al ser muy sencilla y permite seguir día a día el estado de azúcar en la sangre. Tiene una interfaz simple y las nuevas entradas se actualizan de forma rápida.

- **Epocrates:** ésta aplicación sirve para ubicar información acerca de medicamentos, interacción de drogas, identificación de píldoras de acuerdo a sus características, con algunas calculadoras clínicas y otros recursos; también tiene un centro de recursos y tablas por categorías, aunque los restantes requieren de una suscripción paga.

En el inicio de la aplicación se puede buscar los diferentes medicamentos por nombre o por clase; en los siguientes tópicos se requiere suscripción: enfermedades por sistemas, enfermedades infecciosas y exámenes de laboratorio.

En el centro de recursos se encuentra la información acerca de temas tan variados como alergias, asma, fibrilación auricular, desorden bipolar, déficit de atención, EPOC, diabetes, fibromialgia y esquizofrenia.

- **CliniCalc Medical Calculator:** ésta aplicación se utiliza para realizar cálculos con frecuencia de acuerdo a la especialidad, para estimar la severidad de un paciente, clasificar el riesgo de un paciente antes de una cirugía, estimar el déficit de bicarbonato de un paciente o la pérdida de sangre posterior a una hemorragia, etc.

Está clasificada por especialidades y sus cálculos más frecuentemente utilizados, de manera que tiene calculadoras para Anestesia, donde se puede encontrar la clasificación Apache II, la clasificación de la ASA, el déficit de bicarbonato, la clasificación de Cormack, el Score DIC para coagulación intravascular diseminada, el mapa de dermatomas, la equivalencia de opioides, etc.

- **Sana:** ésta aplicación ha sido desarrollada por un grupo de estudiantes del Instituto Tecnológico de Massachusetts (MIT). Este sistema está diseñado para mejorar la prestación de asistencia sanitaria en las zonas rurales y desatendidas mediante código abierto. Sana ha desarrollado una plataforma de código abierto que permite utilizar los teléfonos móviles para capturar y enviar datos de un registro médico electrónico. Vincula a los trabajadores comunitarios de salud con los médicos para apoyar las decisiones en tiempo real.

#### **2.4. Estudio de factibilidad técnica para la implementación de una aplicación para dispositivos móviles**

En el siguiente apartado se describe el análisis realizado en base a la implementación de una aplicación web para dispositivos móviles.

### **2.4.1. Análisis Funcional y Comparativo**

El análisis funcional se realizó por medio de una revisión de dos tipos de aplicaciones para dispositivos móviles que comprendió los siguientes elementos:

- Identificar los principales tipos de aplicaciones para dispositivos móviles, considerando los estándares de comunicación que maneja el sistema actual.
- Definir criterios de comparación y realizar análisis de los tipos de aplicaciones para dispositivos móviles estudiados.
- Obtener conclusiones y sugerencias de uso como consecuencia de la comparación realizada.

El análisis comparativo consistió en evaluar cada tipo de aplicaciones para dispositivos móviles en términos de distintos aspectos, de la forma siguiente:

- Interfaz de usuario.
- Desarrollo.
- Capacidades.
- Accesibilidad.
- Versiones de la aplicación.
- Fortalezas.
- Debilidades.

Basado en estos aspectos se realizó un análisis comparativo que refleja las semejanzas o diferencias en las funciones de cada tipo de aplicación para dispositivos móviles que nos permite valorar sus características principales y crear un criterio claro de selección.

### **2.4.2. Tipos de Aplicaciones para Dispositivos Móviles Analizados**

A continuación se describen los tipos de aplicaciones para dispositivos móviles analizados.

- **Aplicación Móvil Nativa:**

Una aplicación nativa es una aplicación para un determinado dispositivo móvil (*smartphone*, *tablet*, etc). Están instalados directamente en el dispositivo.

Están desarrolladas para cada sistema operativo de forma particular. Los usuarios suelen adquirir estas aplicaciones a través de una tienda en línea o en el mercado, tales como la App Store o aplicaciones Android en Google Play.

- **Aplicación Web Móvil:**

Como lo vimos en el capítulo anterior, las aplicaciones web móviles son aplicaciones móviles que no necesitan ser instaladas o compiladas en el dispositivo de destino. Usando XHTML, CSS y Java Script, son capaces de proporcionar una rica experiencia de la aplicación para el usuario final.

Este tipo de aplicaciones permiten a los usuarios interactuar con el contenido en tiempo real, donde un click o toque realiza una acción dentro de la vista.

### **2.4.3. Análisis Comparativo**

En términos de la apariencia y la sensación general, hay poca diferencia entre las aplicaciones nativas y las aplicaciones web móviles. Generalmente se opta por desarrollar ambas versiones haciendo que la experiencia del usuario sea consistente sin importar desde cuál versión de la aplicación se esté accediendo, por ejemplo Facebook y Twitter.

## **Desarrollo**

A continuación se detallan las características del desarrollo en los diferentes tipos de aplicaciones.

### **Aplicaciones nativas**

- Cada plataforma de desarrollo de aplicaciones móviles (por ejemplo, iOS, Android) requiere su propio proceso de desarrollo.
- Cada plataforma de desarrollo de aplicaciones móviles tiene su propio lenguaje de programación nativo: Java (Android), Objective-C (iOS) y Visual C + + (Windows Mobile), etc.
- Tienen kits de desarrollo de software estándar (SDK). Las herramientas de desarrollo y los elementos comunes de la interfaz de usuario (botones, campos de texto de entrada, etc.) a menudo son proporcionados por el fabricante de la plataforma.

### **Aplicaciones web móviles**

- Se ejecuta en el navegador web del dispositivo móvil y cada uno puede tener sus propias características y peculiaridades. Las aplicaciones web móviles están escritas en HTML5, CSS3, Java Script y lenguajes del lado del servidor elegidos de los desarrolladores (por ejemplo, PHP, Rails, Python).
- Los desarrolladores no tienen que utilizar los kits de desarrollo de software estándar (SDK) para poder construir una aplicación web para móviles.

### **Capacidades**

Según sus capacidades de interacción con el dispositivo, las aplicaciones poseen las siguientes características:

#### **Aplicaciones nativas**

- Pueden interactuar con las características nativas del dispositivo, como lo son: el GPS, la cámara, sonido, explotando las capacidades del hardware del dispositivo.

#### **Aplicaciones web móviles**

- Las aplicaciones móviles web pueden acceder a una cantidad limitada de características nativas del dispositivo y la información.

### **Accesibilidad**

Según el tipo de aplicación a desarrollar la accesibilidad para obtenerla se describe según las siguientes características:

#### **Aplicaciones nativas**

- Descargado en un dispositivo móvil.
- Instalado y se ejecuta como una aplicación independiente (no es necesario un navegador web)
- Los usuarios deben descargar e instalar las actualizaciones de aplicaciones cada vez que se genere una nueva versión.
- Hay tiendas y mercados para ayudar a los usuarios a encontrar su aplicación.

### **Aplicaciones web móviles**

- Acceso a través del navegador web de un dispositivo móvil
- No hay necesidad de instalar software nuevo.
- Las actualizaciones se realizan en el servidor web sin intervención del usuario.
- Puesto que no hay tienda de aplicaciones para la Web móvil, puede ser más difícil para los usuarios encontrar su aplicación.

### **Versiones de la aplicación**

A continuación se detallan las características de versión según el tipo e aplicación móvil.

#### **Aplicaciones nativas**

- Algunos usuarios pueden optar por hacer caso omiso de una actualización, lo que conlleva a que estén ejecutando versiones diferentes de la aplicación.

#### **Aplicaciones web móviles**

- Todos los usuarios están en la misma versión.

### **Fortalezas**

Las fortalezas que posee cada tipo de aplicación móvil son:

#### **Aplicaciones nativas**

- El entorno de desarrollo, mejores prácticas, soporte y kits de desarrollo estándar proporcionados por los fabricantes de dispositivos pueden ayudar a acelerar el desarrollo de las aplicaciones nativas.
- Mejor experiencia del usuario (en la interfaz del móvil) y mejor uso de las características de hardware del mismo.
- Las tiendas de aplicaciones ayudan a los usuarios a encontrar aplicaciones nativas.

#### **Aplicaciones web móviles**

- Contar con una base de código común para todas las plataformas.

- Los usuarios no tienen que ir a una tienda o appStore para descargar la aplicación e instalarla en el dispositivo.
- Costos de desarrollo y producción más económicos unidos al control total sobre la distribución de la aplicación.

### **Debilidades**

A continuación se detallan las debilidades de cada tipo de aplicación:

#### **Aplicaciones nativas**

- Su desarrollo es más costoso, especialmente si se desea desarrollar para múltiples dispositivos móviles, la aplicación debe desarrollarse varias veces, una por cada plataforma que se quiera utilizar (Android, iOS, Blackberry, Windows Phone, etc.). Los usuarios pueden estar en diferentes versiones de la aplicación ocasionando que sea más difícil prestarles soporte.

#### **Aplicaciones web móviles**

- Una interfaz de usuario móvil comparativamente débil y poca capacidad de interactuar con el móvil (sonido, pantalla, GPS, cámara, etc).

### **2.5. Factibilidad técnica para la implementación de una aplicación móvil nativa para dispositivos móviles**

En este apartado se presenta el análisis realizado para la implementación de una aplicación móvil nativa para dispositivos móviles.

#### **2.5.1. Análisis Funcional y Comparativo**

El análisis funcional se realizó por medio de una revisión de los sistemas operativos para dispositivos móviles que comprendió los siguientes elementos:

- Identificar los principales sistemas operativos a estudiar, considerando los estándares de comunicación que maneja el sistema actual.
- Localizar productos en la plataforma de software abierto (Open Source).

- Definir criterios de comparación y realizar análisis de los Sistemas Operativos.
- Obtener conclusiones y sugerencias de uso como consecuencia de la comparación realizada.

El análisis comparativo consistió en evaluar cada producto en términos de distintos aspectos, de la forma siguiente:

- **Punto de vista del Usuario:** consistió en comparar los sistemas operativos desde un punto de vista del usuario final, buscando similitudes o diferencias en sus funciones básicas y algunas de nivel medio o avanzado.
- **Punto de Vista del desarrollador:** consistió en comparar los sistemas operativos desde un punto de vista del desarrollador, tomando en cuenta aspectos relacionados a la facilidad de desarrollo, documentación, costos, capacidades de los equipos entre otras.
- **Penetración del mercado:** consistió en investigar como fue la penetración en el mercado de los distintos sistemas operativos a lo largo de los años pasados y cómo será su evolución en los años siguientes.
- **Funcionalidades:** basado en estos aspectos se crearon tablas que reflejan las semejanzas o diferencias en las funciones de cada sistema operativo para valorar sus características principales y crear un criterio claro de selección.

### 2.5.2. Sistemas Operativos Analizados

Uno de los aspectos que se debe tomar en cuenta cuando se desea desarrollar aplicaciones para dispositivos móviles es sobre cual plataforma o sistema operativo trabajan. Cada una de ellas tiene sus particularidades, no sólo en cuanto al manejo del dispositivo por el usuario, sino también en cuanto a los lenguajes de programación que soportan. [8] Los sistemas operativos analizados fueron:

- Apple iOS.
- Android.
- Windows Phone.
- Blackberry OS.
- Symbian.

### 2.5.3. Análisis Comparativo desde el punto de vista del usuario

Desde el punto de vista del usuario las características de los sistemas operativos móviles son:

#### Apple iOS

- **Aplicaciones:** gran cantidad y de muy buena calidad. A pesar de que hay algunas que están bloqueadas por Apple.
- **Usabilidad y Diseño:** muy fácil de usar, totalmente intuitivo, diseño elegante y una interfaz muy atractiva.
- **Navegación en Internet:** excelente. La función Reader muestra sólo los textos de los artículos, ordenando la información para que sea fácil de leer y seguir, además de reducir la descarga de datos que no nos interesan.
- **Emails y Mensajes:** buena implementación. iMessage permite enviar mensajes, fotos, y videos de manera gratuita entre iPhones, iPods Touch y iPads. Incluso permite chat grupal.
- **Fotos y Videos:** muy buena calidad, hay muchas aplicaciones para mejorar las imágenes o videos que tomamos con el celular. Dispone de un botón para abrir el obturador, que de hecho es muy rápido. Se pueden tomar fotografías con la pantalla bloqueada y las video llamadas funcionan muy bien. Los colores suelen estar muy saturados, lo cual si bien para algunos es algo bueno, para los fotógrafos avanzados esto podría ser un gran problema.
- **Construcción:** la calidad de sus componentes es excepcional. Tanto como su increíble diseño.
- **Batería:** ningún dispositivo Apple tiene batería removible.
- **Memoria:** no existe la posibilidad de añadir memoria, y no se puede utilizar el teléfono como un *Pen-Drive*.

- **Sincronización y Backups:** el servicio iCloud es gratuito para sincronizar y hacer back ups automáticos vía WiFi de nuestros dispositivos Apple, incluyendo fotos, libros, música, documentos, contactos y emails.
- **Personalización:** plataforma cerrada.
- **Actualizaciones:** estandarizadas en la nube.

### Android

- **Aplicaciones:** gran cantidad de aplicaciones, muchas de ellas gratuitas, y sin limitaciones para cualquier proceso de aprobación.
- **Usabilidad y Diseño:** con un poco de trabajo se puede llegar a personalizar completamente, a pesar de seguir siendo un sistema muy complejo. Dispone de comandos de voz integrados.
- **Navegación en Internet:** experiencia de navegación excelente. Soporta Adobe Flash y hasta 16 pestañas simultáneas.
- **Emails y Mensajes:** muy buena implementación, incluye total integración con Gmail. Muy buen reconocimiento de voz. No soporta múltiples cuentas de intercambio de mensajes.
- **Fotos y Videos:** muy buena calidad, tiene opciones como las fotos panorámicas, los videos time-lapse, y editar las fotografías. Obturador muy rápido.
- **Construcción:** incontable variedad de dispositivos disponibles, variando desde construcciones baratas hasta las de la más alta calidad.
- **Batería:** muchos *Smartphones* con este sistema operativo tienen problemas de baja duración de la batería a menos que el usuario modifique algunos parámetros internos.
- **Memoria:** expandible en la mayoría de los teléfonos.

- **Sincronización y Backups:** existen aplicaciones de terceros para sincronizar y hacer backups de nuestra información de manera inalámbrica.
- **Personalización:** plataforma abierta, bajo términos de terceras partes, con al menos casi todo personalizable, aunque debido a la fragmentación del mercado a veces se torna difícil ajustar las personalizaciones y las aplicaciones.
- **Actualizaciones:** al haberse fragmentado tanto el mercado, se tornan un poco lenta y desorganizadas las actualizaciones, ya que no son las mismas para todos los dispositivos.
- **Otros:** Android Beam usa tecnología NFC para compartir contenido solamente con colocar dos Smartphone Android juntos. Puede monitorear la utilización de nuestra información. Aunque existen ciertos riesgos de seguridad debido a la plataforma abierta.

### **Blackberry OS**

- **Aplicaciones:** poca cantidad de aplicaciones disponibles, además de la calidad que deja bastante que desear en la mayoría de los casos, debido a la diferencia de hardware que hay entre los dispositivos Blackberry y el resto.
- **Usabilidad y Diseño:** usable para el usuario en cuanto a llamadas, mensajes y correos electrónicos (*emails*).
- **Navegación en Internet:** buena navegación fluida, soporta Flash, e incluye atajos de teclado.
- **Emails y Mensajes:** la mejor interfaz de emails. Blackberry Messenger viene incluido y permite el intercambio de mensajes en tiempo real de manera gratuita entre equipos Blackberry. La mayoría de los dispositivos trae teclado físico. No soporta múltiples cuentas de intercambio de mensajes.
- **Fotos y Videos:** la calidad de la cámara está muy lejos de la altura de la mayoría de los teléfonos de sus competidores, así como la cantidad de aplicaciones disponibles para mejorarla. La captura y reproducción de videos realmente no es de buena calidad, como

tampoco lo son la pantalla y su resolución para alta definición de imágenes, además que no permite video llamadas.

- **Construcción:** duración y calidad de sus componentes. Tiene un diseño funcional y suele ser menos estético e innovador que el resto de sus competidores.
- **Batería:** siempre Blackberry se ha destacado por su larga duración de baterías, aunque con el nuevo sistema operativo, y pantallas cada vez más grandes, se volvió menos duradera, aunque sigue siendo la mejor del mercado.
- **Memoria:** expandible, buen sistema de administración de memoria.
- **Sincronización y Backups:** la aplicación Blackberry AutoProtect hace un backup automático cada cierto tiempo de contactos, mensajes, calendario y marcadores.
- **Personalización:** una versión intermedia entre Android y iOS.
- **Actualizaciones:** la plataforma está un poco desorganizada y suele tardar un tiempo en ofrecer las actualizaciones para los equipos.
- **Otros:** encriptación de nivel militar con los datos de navegación. Tecnologías NFC están desarrollando la utilización del Blackberry como una especie de tarjeta de crédito para realizar pagos en un comercio.

### Windows Phone

- **Aplicaciones:** la librería de aplicaciones está creciendo, aunque todavía es pequeña.
- **Usabilidad y Diseño:** fácil de usar, interfaz muy fluida e intuitiva. Excelente diseño.
- **Navegación en Internet:** buena experiencia de navegación. Incluye un navegador con compatibilidad full con HTML5. Aunque aún no soporta Adobe Flash, y tiende a tener ciertos errores.
- **Emails y Mensajes:** reconocimiento de voz para emails y mensajes, Text-To-Speech permite que el teléfono nos lea los mensajes.

- **Fotos y Videos:** botones de la cámara físicos. Generalmente los equipos que vienen con este sistema operativo traen pantallas de buen tamaño. Opción Autofix, para mejorar automáticamente las fotos en el momento de la captura.
- **Construcción:** existe una cierta variedad de teléfonos, variando desde baja hasta alta calidad de construcción, ofreciendo distintas características como teclados físicos.
- **Batería:** buena duración de la batería. Ayuda mucho utilizar las opciones de ahorro de energía.
- **Memoria:** expandible en algunos *Smartphones*. No se lo puede utilizar como si fuera un Pen-Drive.
- **Sincronización y Backups:** sincronización automática inalámbrica mediante Zune.
- **Personalización:** pantalla de inicio bastante configurable, interfaz consistente de usuario, con términos de terceras partes, pero no está abierto para personalizar.
- **Actualizaciones:** estandarizadas en la nube.
- **Otros:** integración de Office 365, Word, PowerPoint y Excel.

### Symbian

- Su funcionamiento es óptimo, pero se ha quedado algo obsoleto en lo que a rendimiento multimedia se refiere. En general suele relacionarse con dispositivos menos modernos, que funcionan estupendamente como teléfono móvil, que entretienen pero a los que no les puedes pedir grandes prestaciones porque el sistema operativo no está preparado para ello.
- Tiene el problemas con las aplicaciones. En un mundo en el que las aplicaciones lo son todo, Symbian tiene ese gran defecto, que ha quedado algo apartado por la compañía finlandesa y no se desarrolla nada de última generación.

#### 2.5.4. Análisis Comparativo desde el punto de vista del desarrollador

Cada dispositivo móvil puede tener un sistema operativo particular y se requiere que el desarrollador trabaje con el Kit de Desarrollo de Software (SDK) que es el que provee Interfaz de Programación de Aplicaciones (API), bibliotecas y aplicaciones estándares que permiten crear aplicaciones aprovechando todas las funcionalidades de los dispositivos.

Actualmente existen una gran variedad de tecnologías disponibles para el desarrollo de aplicaciones en dispositivos móviles, específicamente en teléfonos celulares.

##### Apple iOS

- **Lenguajes de Programación:** para implementar las aplicaciones se utiliza el lenguaje Objective-C. Este es un lenguaje orientado a objetos que proviene de C por lo que poseen la misma sintaxis, además de otras características como las sentencias de control de flujo, tipos de datos fundamentales, estructuras y punteros, las funciones y su sintaxis, entre otras.
- **Herramientas:** provee el SDK para el desarrollo de aplicaciones para Mac OS X.
- **Xcode:** es el entorno integrado de desarrollo igualmente para sistemas Mac OS X.
- **Dashcode:** es el entorno de desarrollo integrado que permite desarrollar aplicaciones web para iPhone e iPad, así como mini aplicaciones para Dashboard.
- **Licencia:** para poder desarrollar aplicaciones para iPhone se requiere realizar un registro como desarrollador de Apple y de esta forma obtener los certificados digitales necesarios para que las aplicaciones funcionen en los dispositivos. Este registro tiene un costo asociado. Tanto el SDK como la herramienta de desarrollo Xcode (IDE), son de licencia comercial.
- **Documentación:** hay una extensa documentación oficial que proporciona Apple, que como dueña de la única plataforma de distribución de aplicaciones recauda una parte importante de sus ingresos de las ventas de aplicaciones realizadas en ella con el fin de facilitar la aparición de todo tipo de aplicaciones para su plataforma. Debido también al interés despertado en la comunidad de desarrolladores, los sitios web especializados y

las aportaciones de la comunidad de desarrolladores son extensas, por lo que encontrar librerías y *frameworks* para diversas tareas, así como foros de soporte en Internet, resulta relativamente sencillo.

## Android

- **Lenguajes de Programación:** para crear aplicaciones que trabajen en los dispositivos que poseen esta plataforma existen los siguientes lenguajes:
  - **Java:** es el lenguaje recomendado para esta plataforma y posee una extensa cantidad de clases desarrolladas que permiten explotar todas las funcionalidades de los dispositivos.
  - **C++:** Android posee una serie de librerías escritas en C++ y utilizando Android NDK (Kit de Desarrollo Nativo) se pueden crear partes de aplicaciones en C/C++ que luego son llamadas desde el código Java.
- **Herramientas:** en cuanto a las herramientas de desarrollo hay que destacar que en el caso de Android son multiplataforma (Windows, Mac OS X [Intel] y Linux [i386]). Se proveen kits de desarrollo para varios sistemas operativos que están basados en herramientas libres, como Eclipse.
- **Aplicaciones:** se pueden depurar con el emulador que contiene el SDK antes de instalarlas en los dispositivos.
- **Licencia:** es una plataforma con licencia de software libre y de código abierto por lo que cualquier programador puede crear aplicaciones o extender y modificar el API de Android.
- **Documentación:** la documentación disponible para Android es muy extensa. Esto incluye, tanto a la documentación oficial que cubre todos los aspectos de la plataforma y ofrece diversos ejemplos de código como a la multitud de documentación de terceros disponible públicamente en Internet. Existe además, una gran variedad de libros que abarcan todos los aspectos del desarrollo de aplicaciones para la plataforma y finalmente, un amplio abanico de recursos y sitios web especializados donde encontrar información sobre prácticamente cualquier tema relacionado con el desarrollo de Android. A esto, hay que sumar el apoyo de la comunidad de software libre a la plataforma, por lo que en los principales repositorios de código fuente se puede

encontrar el código fuente completo de programas complejos, lo que supone una gran ventaja a la hora de enfrentarse al desarrollo de nuevas aplicaciones.

### **Blackberry OS**

- **Lenguajes de Programación:** Java es el lenguaje utilizado para programar los dispositivos que poseen esta plataforma.
- **Herramientas:** para el desarrollo de aplicaciones, esta plataforma ofrece el BlackBerry Java SDK el cual provee una variedad de clases en Java que permiten utilizar las funcionalidades del dispositivo.
- **Entorno de desarrollo:** se puede utilizar Eclipse, agregándole un plugin, o el entorno que ofrece el fabricante llamado BlackBerry Java Development Environment (BlackBerry JDE).
- **Licencia:** es una plataforma con licencia de software libre por lo que tanto el SDK como las herramientas de desarrollo pueden descargarse de la página Web oficial sin costo alguno.
- **Documentación:** BlackBerry es una plataforma madura que cuenta con documentación abundante. De forma gratuita se puede acceder a la documentación oficial en el sitio de BlackBerry y, junto con los kits de desarrollo se incluyen algunos ejemplos de código. Los foros oficiales proporcionan también ayuda para la realización de ciertas tareas. Además, la bibliografía existente sobre el desarrollo para BlackBerry es también considerable. Sin embargo, aunque existen sitios especializados con pequeños fragmentos de código para resolver ciertas tareas, es difícil encontrar código fuente de piezas grandes de software en Internet.

### **Windows Phone**

- **Lenguajes de Programación:** los lenguajes comúnmente utilizados para el desarrollo de aplicaciones en esta plataforma son aquellos incluidos dentro del entorno Microsoft Visual Studio:
  - Visual C#: es orientado a objetos y se ejecuta en .NET Framework (incluye las bibliotecas de clases). La sintaxis de C# es muy similar a la de C, C++ o Java.

- Visual Basic: es un lenguaje orientado a objetos en el que se ofrece una gran variedad de elementos de diseño de interfaces de usuario, acceso a datos y con el que las aplicaciones se benefician de la seguridad y la interoperabilidad de lenguajes.

Ambos están incluidos en el entorno de desarrollo de Microsoft, Visual Studio, que está conformado además por otros lenguajes de programación como C#, C++, J#, JScript o ASP .Net, entre otros.

- **Herramientas:** para el desarrollo de aplicaciones en dispositivos que poseen esta plataforma se debe contar con el SDK Windows Mobile que contiene el API y librerías necesarias para acceder a la funcionalidad de los dispositivos.
- **Licencia:** al desarrollar para esta plataforma se tiene una gran cantidad de documentación disponible por parte de Microsoft, sin embargo, la desventaja es que estas herramientas de desarrollo son de licencia comercial por lo que tienen un costo asociado.
- **Documentación:** existe abundante documentación en internet pero no comparable a la de otros sistemas operativos. También proveen documentación, ejemplos de proyectos de aplicaciones, y el emulador que permite hacer pruebas y depurar las aplicaciones.

### Symbian

- **Lenguajes de Programación:** los siguientes lenguajes se utilizan con frecuencia para crear aplicaciones en esta plataforma:
  - C++: ofrece la posibilidad de desarrollar aplicaciones que exploten al máximo las características de hardware de los dispositivos. Ocupan menor tamaño y consumen menor cantidad de memoria, además se ejecutan a mayor velocidad por ser el lenguaje nativo del sistema operativo.
  - Java ME: es un lenguaje orientado a objetos y fue diseñado para ser independiente de la plataforma. Es el entorno de desarrollo más extendido, con mayor número de desarrolladores y aplicaciones disponibles.
- **Herramientas:** adicionalmente en las páginas Web oficiales de los fabricantes se ofrecen los Entornos Integrados de Desarrollo (IDE), para ayudar al programador en el

proceso de edición y depuración del código de las aplicaciones. Entre estos se pueden mencionar:

- **Licencia:** una de las ventajas que ofrece a los programadores es que tanto el SDK como las herramientas de desarrollo son de licencia libre y su distribución es gratuita.

La tabla 1 [34] se muestran las características técnicas que tienen los Sistemas Operativos móviles estudiados desde el punto de vista del desarrollador.

**Tabla 1. Características técnicas que tienen los Sistemas Operativos Móviles**

| Sistema Operativo           | Apple iOS            | Android         | Windows Phone          | Blackberry OS     | Symbian             |
|-----------------------------|----------------------|-----------------|------------------------|-------------------|---------------------|
| Núcleo del SO               | Mac OS               | Linux           | Windows CE 7           | BlackBerry OS     | EKA2                |
| Arquitecturas (CPU)         | ARM                  | ARM, MIPS, x86  | ARM                    | ARM               | ARM                 |
| Lenguajes de Programación   | Objective -C, C, C++ | Java, C, C++    | .NET (Silverlight/XNA) | Java              | C++                 |
| IDE                         | Xcode                | Eclipse         | Visual Studio          | Eclipse / JDE     | QtCreator           |
| Disponibilidad de depurador | Si                   | Si              | Si                     | Si                | Si                  |
| Plataforma de desarrollo    | Mac OS X             | Multiplataforma | Windows                | Windows/ Mac OS X | Multiplataforma     |
| Licencia                    | Propietaria          | Libre           | Propietaria            | Propietaria       | Libre / Propietaria |
| Motor de Navegador Web      | Webkit               | Webkit          | Pocket IE              | Webkit            | Webkit              |
| Multitarea                  | Si                   | Si              | Si                     | Si                | Si                  |
| Documentación               | Buena                | Muy Buena       | Regular                | Buena             | Buena               |

### 2.5.5. Análisis Comparativo desde el punto de vista de la penetración del mercado

Otro aspecto fundamental a la hora de comparar las plataformas móviles es su cuota de mercado. En la figura 13 podemos ver un estudio donde se muestra la evolución del mercado de los sistemas operativos móviles según el número de terminales vendidos. Podemos destacar el importante descenso de las ventas de Symbian, además del ascenso continuado de la plataforma Android.

Este estudio es realizado por Gartner Group [34], donde es evidente el crecimiento que ha tenido el sistema operativo Android en cuanto a penetración en el mercado, y se espera que en los próximos 3 años sea el sistema operativo con mayor presencia en los dispositivos móviles acumulando un 50% de uso, seguido muy de cerca por Windows Phone y el iOS quienes tendrían un 40% de presencia (20% del mercado cada uno) dejando en último lugar a BlackberryOS con tan solo un 10%.

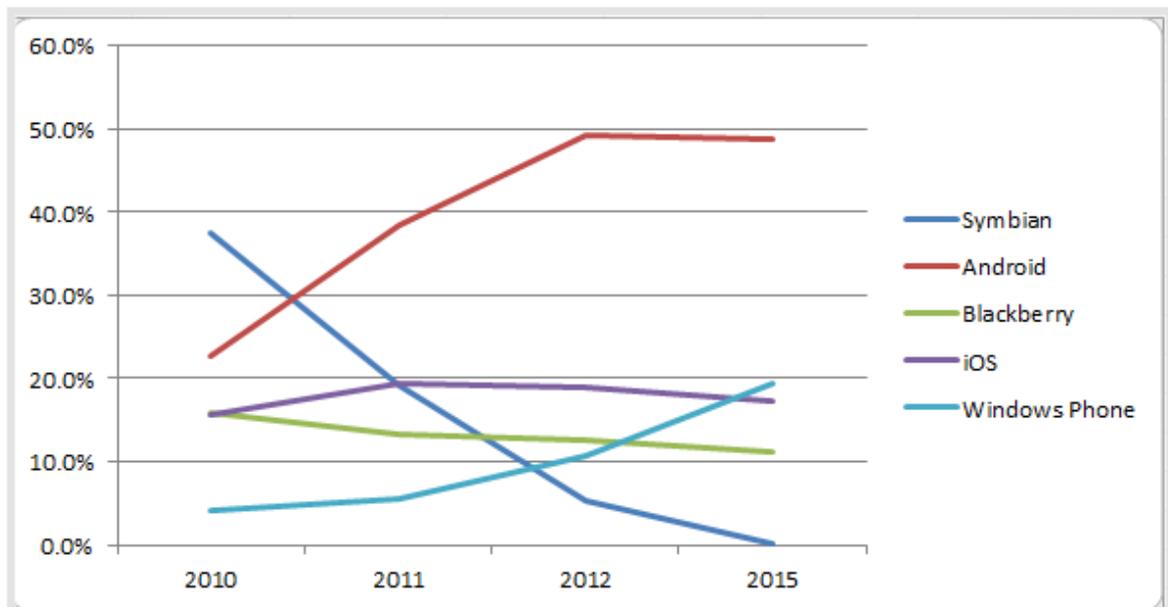


Figura 13. Evolución de los diferentes Sistemas Operativos y su penetración en el mercado

Fuente: <http://www.androiddevmx.net>

## 2.6. Telemedicina

La manera más sencilla de definir Telemedicina es como el uso de tecnologías de comunicaciones para ofrecer servicios e información médica. Según esta definición, la

Telemedicina puede ser tan fácil como una simple llamada telefónica entre dos profesionales de la salud y tan compleja como una transmisión de una videoconferencia usando un enlace de señal satelital entre dos médicos en diferentes ciudades. Sin embargo, si se realiza un estudio de las experiencias en Telemedicina que se han llevado a cabo alrededor del mundo, se verá que en ocasiones no es posible explicar qué se considera Telemedicina y qué queda fuera de este concepto, debido a esto algunos autores creen que Telemedicina tiene un concepto más amplio y que sus límites no están claramente definidos. Por esta razón se van a presentar diferentes conceptos de algunas instituciones y autores especializados en el área [1]:

*“Telemedicina es el suministro de servicios de atención sanitaria, en los que la distancia constituye un factor crítico, por profesionales que apelan a las tecnologías de la información y de la comunicación con objeto de intercambiar datos para hacer diagnósticos, recomendar tratamientos y prevenir enfermedades, así como para la formación permanente de los profesionales de atención de salud y en actividades de investigación y evaluación, con el fin de mejorar la salud de las personas y de las comunidades en que viven”.*

Organización Mundial de la Salud (OMS)

*“Telemedicina es el intercambio de información médica de un sitio a otro usando comunicaciones electrónicas, con el fin de brindar cuidado y educación al paciente o apoyar a los profesionales de la salud con el propósito de mejorar el cuidado del paciente”.*

American Telemedicine Association

*“Telemedicina es el uso de las tecnologías de la información y de las comunicaciones como un medio de proveer servicios médicos, independientemente de la localización tanto de los que ofrecen el servicio, los pacientes que lo reciben, y la información necesaria para la actividad asistencial”.*

INSALUD - Ministerio de Sanidad y Consumo. España

En cada una de estas definiciones se puede encontrar un factor en común, el cual constituye la base sobre la cual se apoya la Telemedicina: las tecnologías de información y comunicaciones.

La Telemedicina utiliza las tecnologías de información y comunicaciones para transferir información médica con el fin de realizar diagnósticos, terapias o entregar educación a distancia. La información que se transfiere puede incluir imágenes, audio o video en vivo,

registros médicos de pacientes, así como también cualquier tipo de datos proveniente de dispositivos médicos. [1]

### 2.6.1. Servicios de la Telemedicina

El término Telemedicina puede incluir una variedad de servicios, los cuales se categorizan de la siguiente manera [1]:

#### 2.6.1.1. Asistencia remota:

La asistencia remota consiste en la utilización de sistemas de información y telecomunicaciones para proporcionar asistencia médica a distancia. Dentro de esta categoría se puede distinguir entre:

- **Consulta / diagnóstico:** consiste en la capacidad de realizar consultas o solicitar una segunda opinión de manera remota entre diferentes facultativos para la elaboración de un diagnóstico común. Entre las principales aplicaciones de la telemedicina a los procesos asistenciales podría incluirse la teleconsulta o telediagnóstico, la cual ofrece aplicaciones para diferentes especialidades médicas:
  - **Radiología:** consiste en la transmisión de imágenes radiológicas a través de redes de comunicación. Se incluye la radiografía convencional, la tomografía axial computarizada, la resonancia magnética, ultrasonidos, etc. Este servicio permite la consulta y el diagnóstico compartido para exploraciones radiológicas.
  - **Cirugía:** permite la realización de intervenciones quirúrgicas por medio de robots manejados de forma remota.
  - **Dermatología:** consiste en el diagnóstico y el tratamiento clínico de problemas dermatológicos a distancia.
  - **Cardiología:** utiliza las tecnologías de la información para la transmisión de electrocardiograma, ecocardiogramas, estudios hemodinámicos etc.
- **Monitorización / vigilancia:** se basa en la posibilidad de realizar un seguimiento a distancia de parámetros relacionados con un proceso asistencial (registro

electrocardiográfico en pacientes con dolor torácico, parámetros vitales en accidentados, etc.), o bien, seguir a distancia la evolución de pacientes crónicos.

Se puede emplear Telemedicina como medio de controlar a distancia la situación del paciente y diferentes funciones vitales. Puede realizarse desde el domicilio del paciente para el seguimiento de enfermos crónicos y procesos postoperatorios, como apoyo a los programas de atención domiciliaria y cuidados paliativos, o en el entorno de la medicina de urgencias mediante el envío de parámetros vitales desde el lugar donde se encuentre el paciente al hospital.

La tecnología actual permite a los usuarios pedir ayuda en una situación de emergencia, activándose un servicio de asistencia a domicilio o la puesta en marcha de un procedimiento de urgencia.

La vigilancia remota asociada a la monitorización permite, por ejemplo, realizar un programa de rehabilitación desde el domicilio bien mediante el acceso a un protocolo de rehabilitación a ejecutar por el paciente o bien mediante la conexión audiovisual con un rehabilitador para la realización de la sesión.

## **2.7. SOS Telemedicina para Venezuela**

SOS Telemedicina para Venezuela es un programa de la Facultad de Medicina de la Universidad Central de Venezuela, dirigido a apoyar con conocimiento experto, a distancia y con el uso de las herramientas que ofrece la tecnología, a profesionales y estudiantes de la salud que se desempeñan en centros de atención primaria. [1]

SOS Telemedicina para Venezuela reduce las barreras en el acceso a los servicios asistenciales, en beneficio directo de pacientes que no cuentan con fácil acceso a una consulta con médicos especialistas. [1]

### **2.7.1. Triage**

Contreras J., Restrepo J. y Múnera A. (2006), indican de triaje [2]:

*“El triaje es un término de origen francés que significa seleccionar, escoger o priorizar; fue utilizado inicialmente en la primera guerra mundial en la atención del gran número de heridos en combate. “*

*“Se entiende por triaje el proceso de categorización de lesionados basado en la urgencia de sus lesiones y la posibilidad de supervivencia; diferente al criterio de atención en condiciones normales, en las que el lesionado más grave tienen prioridad sin tener en cuenta en pronóstico inmediato o a largo plazo.”*

Tomando en consideración los conceptos anteriores, se puede definir triaje como el proceso de selección y clasificación de los pacientes que ingresan a un servicio clínico, donde se le debe brindar una valoración de la enfermedad que presenta y la atención que requiere antes de que reciban la atención asistencial que precisan, garantizándole al paciente la atención dentro del tiempo requerido según su prioridad y disposición del área de urgencias.

### **2.7.2. Referencia Médica**

Ministerio de Salud de Venezuela (1978), indica de referencia médica [2]:

*“Es el proceso por medio del cual se envía a un paciente para su atención de un nivel a otro, dentro del mismo nivel, o de una institución a otra con mayores recursos para el diagnóstico y tratamiento.”*

Vargas Ileana y Castillo Alcira (2002), agregan a la definición:

*“Según sea el padecimiento, el especialista asumirá el manejo del paciente durante todo el curso de la enfermedad, parte de ella; posteriormente el paciente es enviado o devuelto al referente.”*

En base a los conceptos anteriores, se puede definir una referencia médica como el procedimiento que se utiliza para conducir a un paciente de alguna unidad médica a otra unidad que tenga una mayor capacidad para resolver el inconveniente clínico, con el fin de brindarle al paciente la atención médica correcta.

## **2.8. Antecedentes**

En éste apartado se muestran los antecedentes y las investigaciones realizadas que sirvieron como base para el desarrollo de este estudio.

**2.8.1. “Sistema de Historia Médica Electrónica para SOS Telemedicina”.** El objetivo de este trabajo fue analizar e implantar un sistema de historias médicas electrónicas aplicando estándares internacionales para SOS Telemedicina, lo que ayuda a comprender cuales son las necesidades del proceso asistencial llevado a cabo en centros de salud y a identificar cuáles son las necesidades primordiales para el manejo de los datos clínicos en dichos centros.[1]

SOS Telemedicina para Venezuela es un proyecto que contempla el diseño, desarrollo e implementación de las soluciones tecnológicas que permitan acercar oportunamente servicios especializados de salud de calidad y sin salir de su región a la población, en particular en zonas remotas del país. Es gestionado por el Centro de Análisis de Imágenes Biomédicas Computarizadas (CAIBCO) perteneciente a la Instituto de Medicina Tropical, Facultad de Medicina, Universidad Central de Venezuela.

En este sentido, el programa SOS Telemedicina para Venezuela tiene como objetivo, entre otros, establecer una red de telemedicina a nivel nacional. Los objetivos específicos de esta investigación están orientados hacia la recopilación de la información, que permita modelar el dominio clínico en el ámbito de una historia médica integral y materno infantil, y así evaluar las herramientas software existentes para poder determinar si cumplen con los estándares internacionales para el diseño, comunicación y terminología de los datos clínicos.

De esta investigación se pudo comprender que la mayoría de los sistemas desarrollados bajo un contexto clínico particular, corren el riesgo de ser incapaces de cambiar y/o adaptarse, siendo inservibles fuera del contexto original en el cual fueron creados, y por eso es necesario aplicar estándares de diseño que permitan obtener como resultado un sistema de historias médicas electrónicas estandarizado que permita el intercambio de información entre varios centros de salud, y más aún intercambiar la información entre varios sistemas.

También se comprendió que intercambiar o difundir una información clínica tiene diversos fines, entre ellos: estadísticos, educativos, investigación, logísticos, etc. Pero lo más importante del intercambio de información radica en que beneficia directamente al paciente, pues ayuda a que su expediente de historia médica sea único, es decir sin duplicados, gracias a la integración de distintos expedientes.

**2.8.2. “Desarrollo de un Sistema Web de Telemedicina para proveer Segunda Opinión Médica Especializada”.** Este trabajo tuvo por objetivo principal el desarrollo de un sistema Web de Telemedicina que permita a los médicos o estudiantes de medicina en ambulatorios rurales, obtener a través de Internet segundas opiniones médicas de especialistas de manera eficiente y sencilla. Conjuntamente a la posibilidad de obtener una segunda opinión médica especializada, el sistema debe ofrecer a los médicos rurales y estudiantes contenidos de valor en atención primaria en salud y proveer una herramienta que permita agilizar el acceso a la información de los pacientes (historias médicas, resultados de laboratorio, entre otros). [10]

Este Trabajo Especial de Grado tuvo como caso de estudio la implantación de un sistema de Telemedicina Web para proveer segundas opiniones médicas especializadas a profesionales de la salud y pasantes en zonas rurales.

Gracias al análisis realizado en esta investigación se pudo detectar que en Venezuela, al igual que en el resto de países del mundo existen poblaciones rurales, las cuales se encuentran dispersas o alejadas de los principales centros urbanos. Por esta razón y debido a la centralización de los recursos del país, dichos poblados poseen un nivel de calidad de vida bajo, en el cual, sus habitantes tienen dificultades para acceder a servicios básicos como vialidad, comunicaciones, educación, salud, y es por esto que la asistencia médica en las zonas rurales del país se lleva a cabo en ambulatorios rurales, los cuales están equipados con los instrumentos básicos requeridos para brindar el servicio médico a la población. En cuanto al personal que asiste dentro de un ambulatorio rural, mayormente se trata de médicos y de estudiantes de medicina que están en el internado rotatorio.

Se pudo determinar con la investigación que la mayor cantidad de conocimiento experto o experiencia especializada queda concentrada en los principales centros urbanos de nuestro país, y es por eso que resulta ideal el uso de tecnologías de telecomunicaciones y de computación para mejorar la asistencia médica en las zonas rurales, permitiendo a los médicos y pasantes de los ambulatorios rurales transferir información médica sobre un caso clínico, con el fin de consultar a médicos especialistas y obtener una segunda opinión médica de manera eficiente, sencilla y económica.

**2.8.3. “Apoyo Tecnológico en el diseño, desarrollo, instalación y configuración de aplicaciones, Programa SOS Telemedicina para Venezuela”**, uno de los objetivos generales de ésta investigación fue el desarrollo de un módulo Web para el Sistema SOS Historia Médica Electrónica, que permita gestionar las referencias de la historia médica integral de un paciente.[2]

Con la realización de este Trabajo Instrumental de Grado se logró la creación de soluciones, además de configuración y migración de material electrónico de salud.

La creación del Sistema SOS-TRIAJE que recibe casos clínicos enviados desde el Sistema SOS HME, pretende brindar un mejor funcionamiento a los centros asistenciales pertenecientes al Programa SOS Telemedicina en cuanto a referencias médicas se refiere, debido a que ofrece la posibilidad de alcanzar un mayor avance en los resultados de diagnósticos emitidos al permitir al alcance de un clic, solicitar apoyo para diagnosticar a un paciente mediante la solicitud de una referencia medica a especialistas que se encuentran en un lugar distante al del centro asistencial; ayuda de igual manera al paciente al evitarle la engorrosa y necesaria tarea de tener que trasladarse a algún otro centro asistencial en caso de que se necesite alguna otra opinión o en caso de que en dicho centro no se cuente con los materiales para diagnosticarlo; con la posibilidad de cargar imágenes y archivos puede remitirse un paciente y recibir una referencia medica y evitar el traslado del paciente. Finalmente, este sistema Web generará mejoras importantes en el manejo actual de las referencias médicas.

## CAPÍTULO III – MARCO METODOLÓGICO

### 3.1. Introducción

En este capítulo se presenta la metodología del proyecto conformado por las diferentes etapas de desarrollo de software de la investigación, las técnicas y procedimientos que son utilizados para llevar a cabo la indagación. Se presenta el "cómo" se realizó el estudio para responder al problema planteado.

### 3.2. Metodología de desarrollo

Hacer uso de una metodología ágil promueve el desarrollo de iteraciones a lo largo de la vida del proyecto. Por lo general al final de cada iteración se cuenta con una versión funcional del sistema. [13] En esta investigación se utilizó una adaptación de la metodología de desarrollo «Programación Extrema, *Extreme Programming*», mejor conocido como «XP».

La metodología XP pone el énfasis en la adaptabilidad del desarrollo más que sobre la previsibilidad. En la metodología XP se considera que los cambios de los requisitos sobre la marcha son un aspecto natural, inevitable y en muchos casos indeseable. Por tanto ser capaz de adaptarse a los posibles cambios de los requisitos durante la vida del proyecto es una aproximación que intenta definir todos los requisitos para luego invertir esfuerzos en controlar que no cambien. [14]

Las características fundamentales de la programación extrema son: simplicidad, comunicación, retroalimentación y coraje. [14]

- **Simplicidad:** es la base de la programación extrema. Dividir el problema en pequeñas partes alcanzables y a medida que el proyecto crece se deben redefinir las partes para evitar que la complejidad aumente en forma desmedida. La programación extrema apuesta que es más sencillo hacer algo simple y tener trabajo extra para cambiarlo si se requiere, que realizar algo complicado y nunca utilizarlo.
- **Comunicación:** debe ser constante entre programadores y fluida con el cliente. El cliente forma parte del equipo de desarrollo. Se considera que el código autodocumentado es más fiable y debe documentarse solo aquello que no va a variar, por ejemplo el objetivo de una clase o un método.

- **Retroalimentación:** se realiza un reajuste del código y sus interfaces al final de cada ciclo iterativo, de manera de reorientar el proyecto en base a satisfacer los requisitos restantes y/o realizar modificaciones sobre partes desarrolladas que no alcanzan a los requisitos planteados. La opinión del cliente es muy importante y se conoce en tiempo real.
- **Coraje:** le permite a los desarrolladores sentirse cómodos para construir el código cuando sea necesario. Esto significa revisar el estado del sistema existente y realizar cambios sobre el código creado, si estos cambios a futuro repercuten en una implementación más simple. Un ejemplo tangible de coraje es saber cuándo desechar código obsoleto, sin importar el tiempo y el esfuerzo invertido en crearlo.

Las fases de la metodología XP son (ver figura 14):

- 1) **Planificación:** en esta primera fase se debe hacer primero una recopilación de todos los requerimientos del proyecto, también debe haber una interacción con el usuario, y se debe planificar bien entre los desarrolladores del proyecto que es lo que se quiere para el proyecto para así lograr los objetivos finales.
- 2) **Diseño:** se sugiere que hay que conseguir diseños simples y sencillos. Para procurar hacerlo todo lo menos complicado posible para el usuario o cliente, para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo para desarrollarlo. En esta fase se logrará crear parte del proyecto la parte física (lo bonito) la interfaz que tendrá el usuario o cliente con el proyecto.
- 3) **Codificación:** el cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de X.P. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada. En esta fase de la codificación los clientes y los desarrolladores del proyecto deben estar en comunicación para que los desarrolladores puedan codificar todo lo necesario para el proyecto que se requiere, en esta fase está incluido todo lo de codificación o programación por parte de los desarrolladores del proyecto.

- 4) **Pruebas:** uno de los pilares de la metodología X.P es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando. Para esta fase lo que se implementa es el uso de test que son pruebas que se le hacen al proyecto o como ya se dijo a los códigos que se vayan implementando.



Figura 14. Etapas De La Metodología XP

### 3.3. Etapas de desarrollo

Una vez realizado el estudio de la metodología a aplicar, se consideraron las siguientes etapas desarrollo de software de ésta investigación.

#### 3.3.1. Etapa I. Planificación

En esta primera etapa se definió el proceso para identificar las necesidades existentes, a través de las diferentes reuniones con el equipo del Centro de Informática Médica y de los estudios realizados, se plantearon las soluciones posibles con el propósito de definir y destilar los requerimientos para cumplir las restricciones impuestas por las distintas partes.

Esta etapa tiene un gran nivel de complejidad dado que de ella depende una gran parte del éxito del sistema. Por ello, para realizar un buen proceso de determinación de requerimientos, el desarrollo debió soportar de la especificación de cada uno de los requerimientos planteados, de tal forma que tengan los siguientes atributos:

- Deben ser completos, consistentes y han de estar dentro del alcance del proyecto.
- Deben contemplar el esquema de flujo del sistema SOS-TRIAJE existente.
- Cumplir con los objetivos del Centro de Informática Médica.
- Son viables y apropiados para el desarrollo.
- Los requerimientos han de ser "testeables" (deben tener capacidad de prueba).

Una vez definidos y estudiados con profundidad los requerimientos, se comienza a estructurar el diseño a través del cual se pondrá en marcha el sistema.

### **3.3.2. Etapa II. Diseño**

En esta etapa se desarrolló el análisis y estudio del diseño mediante el cual es implementado el sistema móvil.

Se definió la interfaz del sistema, así como los modelos del aspecto gráfico, del contenido, y del funcionamiento del mismo, en base a los requerimientos, necesidades planteadas y formatos existentes en las aplicaciones propias del Programa SOS Telemedicina para Venezuela.

Los modelos desarrollados son prototipos del sistema y lo reflejan con exactitud, así como son acordes a los colores, formas y diseño de los sistemas ya existentes, para poder ir evolucionándolos hasta que queden satisfechos con todos los aspectos.

El diseño y el análisis se implementó a través de diferentes diagramas UML (Lenguaje Unificado de Modelado, *Unified Modeling Language*) como lo son: diagrama de casos de uso, diagrama de secuencia, diagrama de clases y diagrama entidad – relación, a través de los mismos se pudo conocer a fondo el funcionamiento del sistema, para así desarrollar las diferentes funciones acordes al sistema ya existente.

### **3.3.3. Etapa III. Codificación**

Mediante esta etapa se realizó el modelo de diseño, es decir se elaboran y se adaptan los elementos gráficos y multimedia, se codifican las páginas, los programas y scripts, se definen y preparan las bases de datos para que el sitio quede en funcionamiento. Se aplica el diseño para ambas aplicaciones en desarrollo (Web móvil y móvil nativa), se codifica en los lenguajes propios de cada una y se pone en funcionamiento.

El desarrollo se implementó bajo una arquitectura cliente-servidor, ver figura 15. En la arquitectura, la capacidad de proceso está repartida entre los clientes y los servidores, Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

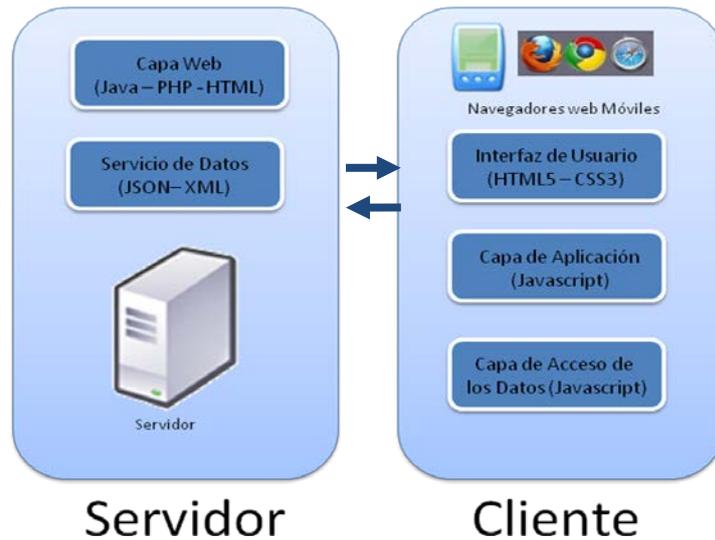


Figura 15. Arquitectura del sistema móvil de Referencias Médicas

Se utilizan patrones de diseño. En particular el patrón de diseño «Modelo Vista Controlador», conocido por sus siglas como «MVC». [17]. Es un patrón o modelo de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. Es usado en aplicaciones web, dónde la vista es la página HTML, el modelo es el Sistema de Gestión de Base de Datos y la lógica interna, y el controlador es el responsable de recibir los eventos y darles solución. La finalidad del modelo es mejorar la reusabilidad por medio del desacople entre la vista y el modelo. Los elementos del patrón son los siguientes [17]:

- **Modelo:** es la representación de la información en el sistema. Trabaja junto a la vista para mostrar la información al usuario y es accedido por el controlador para añadir, eliminar, consultar o actualizar datos.
- **Vista:** es la presenta al modelo en un formato adecuado para que el usuario pueda interactuar con él, casi siempre es la interfaz de usuario.
- **Controlador:** es el elemento más abstracto. Recibe, trata y responde los eventos enviados por el usuario o por la propia aplicación. Interactúa tanto con el modelo como con la vista.

Con la utilización de este modelo es posible tener diferentes vistas para un mismo modelo (ejemplo representación de un conjunto de datos como una tabla o como un diagrama de barras). [17] Es posible construir nuevas vistas sin necesidad de modificar el modelo subyacente.

Proporciona un mecanismo de configuración a componentes complejos muchos más tratable que el puramente basado en eventos (el modelo puede verse como una representación estructurada del estado de la interacción), como se muestra en la figura 16.

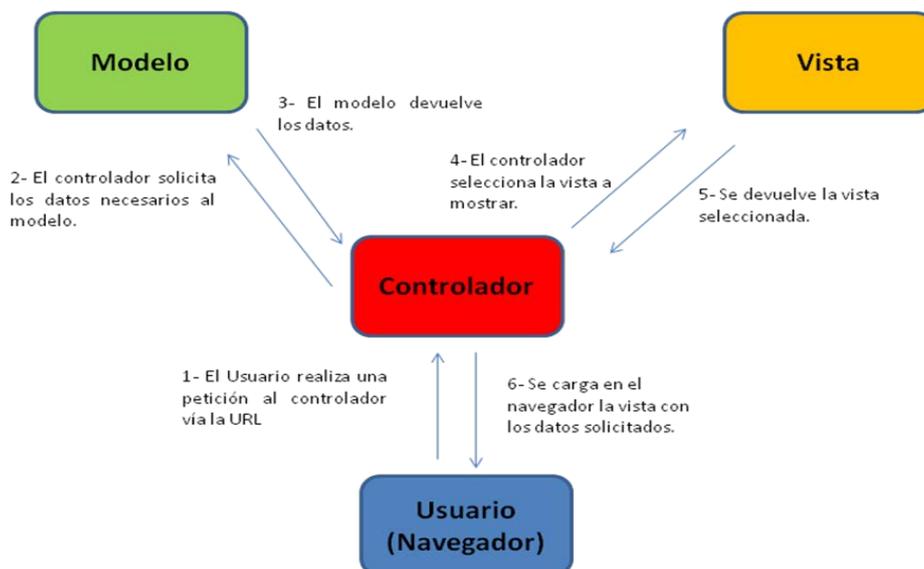


Figura 16. Flujo de interacción de MVC

Luego de esta fase el sistema está listo para comenzar a funcionar.

#### 3.3.4. Etapa IV. Pruebas

Una vez que se han desarrollado los sistemas en base al diseño establecido, se habilitan, en Internet, mediante los servidores locales y en las bases de datos locales, para que los usuarios comiencen a utilizar los mismos, y mediante éstos detectar posibles fallas o errores para su posterior corrección.

En esta etapa se realizan las pruebas exhaustivas para asegurar el perfecto funcionamiento del mismo, se prueba la integración con los sistemas internos, todo esto se hace primero en el ambiente de desarrollo y luego en el servidor de Internet en el que realmente funcionará.

Ahora los sistemas se encuentran en pleno funcionamiento con usuarios accediendo y con los administradores realizando las tareas periódicas de mantenimiento, brindando atención a los usuarios, actualización de contenidos y permitiendo el desarrollo de pruebas funcionales y de aceptación por parte del usuario final.

## **CAPÍTULO IV – MARCO APLICATIVO**

### **4.1. Introducción**

Este capítulo contiene la descripción detallada de las soluciones móviles de software desarrolladas, y se realiza un acercamiento sobre el contexto de éstas soluciones. Además se exponen las tecnologías utilizadas en el desarrollo de la aplicación, detallando los componentes de la plataforma de desarrollo. El desarrollo está basado en tecnologías HTML y Java.

En este capítulo hace referencia a las actividades realizadas y las decisiones tomadas durante el desarrollo. De las aplicaciones de software desarrolladas, se describe la arquitectura, el diseño y los detalles de implementación que se consideraron pertinentes exponer.

Finalmente se muestran el conjunto de pruebas realizadas sobre el producto final y los resultados obtenidos.

### **4.2. Contexto del proyecto**

Este proyecto se enmarca dentro del programa SOS Telemedicina para Venezuela, específicamente en el Sistema de Referencia Médica (SOS-TRIAJE) y ha sido desarrollado en el contexto del Trabajo Especial de Grado presentado ante la Universidad Central de Venezuela, Facultad de Ciencias, Escuela de Computación, bajo la tutoría de la Profesora Ana Morales, para optar al Título de Licenciado en Computación, por parte de los bachilleres: Vanessa Rodríguez y Alberto Bay, como integrantes del equipo desarrollador.

El desarrollo fue supervisado por el equipo del Centro de Informática Médica así como fueron quienes coordinaron la toma de decisiones y realizaron pruebas.

### **4.3. Detalle de la solución**

Se realizaron un conjunto de reuniones con los diferentes profesionales presentes en el Programa SOS Telemedicina para Venezuela, a partir de las mismas se identificó que:

1. Entre las soluciones desarrolladas en el programa se encuentran: SOS-HME: Un sistema de historias médicas electrónicas, basado en estándares internacionales, que permite registrar información clínica de pacientes en centros asistenciales, remitir pacientes a

uno o varios especialistas e intercambiar información clínica con otros sistemas; y SOS-TRIAJE: Un sistema Web, que permite gestionar las referencias de la historia médica integral de un paciente, haciendo uso de su historia médica y demás datos del paciente para que médicos especialistas emitan impresiones diagnósticas o segundas opiniones de salud, acerca de diferentes patologías presentadas.

2. La necesidad que existía de implementar el Sistema de Referencias Médicas, a través de dispositivos móviles, y más específicamente en “*smartphones*”.

Para conocer los requerimientos y especificaciones del nuevo sistema, se estudiaron los proyectos que dieron vida a los sistemas mencionados, a saber: “Sistema de Historia Médica Electrónica para SOS Telemedicina” y “Apoyo tecnológico en el diseño, desarrollo, instalación y configuración de aplicaciones, Programa SOS Telemedicina para Venezuela”, se exploraron los sistemas, su funcionamiento, los módulos que los conforman y la interrelación entre ambos. A partir de estas actividades se proyectó el alcance de implementar SOS-TRIAJE para dispositivos móviles.

Los requerimientos funcionales y no funcionales definidos son los siguientes:

### **Requerimientos funcionales**

Se requiere desarrollar un sistema móvil de Referencias Médicas, que cumpla con los siguientes requerimientos funcionales:

- **Requerimiento 1 : autenticación y autorización:**

La aplicación web móvil, deberá poseer tres tipos de usuarios, el usuario «Médico Rural» (exclusivamente médicos rurales), el usuario «triaje» (para médico y personal capacitado en la unidad de «Médico Triage»), y el usuario «Especialista» (médicos especialistas que emiten segundas opiniones). Por su parte la aplicación nativa en Andorid deberá contemplar únicamente al usuario «Médico Rural».

Todos los usuarios deberán autenticarse para poder ingresar al sistema. Solo el personal médico y sanitario tendrá acceso al registro clínico.

- **Requerimiento 2: aceptación o rechazo de un caso:**

El usuario «Médico Especialista», una vez autenticado y dentro de sus sesión correspondiente, podrá aceptar o rechazar casos de solicitud de referencia médica, que les sean asignados en un momento determinado. Si el usuario rechaza uno de los casos el usuario «Médico Triage» será el encargado de responderlo o reasignar el mismo.

- **Requerimiento 3: envío y recepción de información clínica:**

Los usuarios pueden enviar y recibir la información clínica asociada a un paciente, a través de texto y contenido multimedia. El envío de la información asociada a un paciente es la función principal del sistema permite que los diferentes usuarios se comuniquen a través del sistema de forma narrativa y que puedan describir cualquier fenómeno o patología que presenta un determinado paciente, pero también ha de ser estructurado para que la información clínica pueda ser computada.

- **Requerimiento 4: resumen de información clínica:**

La información clínica correspondiente a un paciente debe poder ser vista en el Sistema de Referencias Médica (SOS-TRIAJE) por pantalla al igual que desde el Sistema Móvil de Referencia Médicas desde los dispositivos móviles de los usuarios.

- **Requerimiento 5: no dependencia de Internet**

La aplicación nativa en Android no debe depender de conexión a Internet para su funcionamiento, debido a que ha de ser instalado en zonas rurales de baja conectividad. Por esto se requiere que el sistema funcione bajo un esquema «stand alone».

Sin embargo es recomendable que permita conectividad en un momento determinado para poder sincronizar la información con el Sistema de Referencias Médicas (SOS-TRIAJE).

- **Requerimiento 6: Creación y cierre de un caso**

El usuario «Médico Rural» será el único capaz de crear un nuevo registro de un caso de segunda opinión en salud así como de cerrarlo, bien sea cuando se obtuvo la segunda opinión solicitada o cuando lo considere apropiado.

- **Requerimientos 7: Emisión de segundas opiniones**

Los usuarios «Médico Triage» y «Médico Especialista» podrán emitir una opinión con respecto a un determinado caso para lograr la solución del mismo.

- **Requerimiento 8: Asignar / Reasignar caso**

El usuario «Médico Triage» será el único capaz de asignar ó reasignar un caso a un determinado médico especialista.

### **Requerimientos no funcionales**

El sistema móvil de Referencias Medicas debe cumplir con los siguientes requerimientos no funcionales:

- **El sistema de ser usable:** su interfaz gráfica debe ser agradable a la vista, intuitiva e ir acorde con el la imagen del proyecto SOS Telemedicina y el módulo de Referencias Medicas.
- **El sistema debe ser escalable:** de manera que pueda extender su margen de operaciones permitiendo la agregación de nuevas funcionalidades.
- **El sistema debe ser mantenible:** es decir que pueda ser modificado para corregir fallas y mejorar su funcionamiento.
- **El sistema debe de ser robusto:** pudiendo recuperarse de posibles errores y mantenerse en funcionamiento.
- **Utilizar tecnologías no propietarias:** el sistema debe ser realizado bajo un entorno de desarrollo *open source*.

La razón principal de desarrollar un sistema móvil que se integre al Sistema SOS-TRIAJE, es facilitar el acceso a una segunda opinión en salud emitida por especialistas y, requerida por los profesionales de los centros asistenciales, haciendo uso de sus dispositivos móviles y sin importar el lugar en donde se encuentren.

A partir de las necesidades planteadas y del estudio de los sistemas: SOS-HME y SOS-TRIAJE, se llevó a cabo un estudio de factibilidad para el desarrollo de una aplicación Web móvil y una aplicación móvil nativa. Se realizaron análisis comparativos entre las ventajas y desventajas de ambas, así como investigaciones acerca del desarrollo de aplicaciones existentes y su utilidad en

el ámbito de la medicina actual. Se estudiaron los diferentes dispositivos presentes en el mercado, sus sistemas operativos, el grado de demanda de los mismos y las ventajas que ofrece su desarrollo. El resultado de este estudio demostró:

1. Una aplicación web móvil puede ser utilizada en todos los modelos de dispositivos móviles, pero se pierden los beneficios adicionales que ofrecen algunas tecnologías.
2. Una aplicación móvil nativa incorpora a los sistemas beneficios propios de la tecnología ofrecida por el dispositivo, requiriéndose la realización de una aplicación nativa para cada uno de los modelos, así como las acciones gerenciales impuestas por las casas proveedoras de cada tecnología.
3. De las 5 tecnologías estudiadas, Ver Tabla 1, la de mayor demanda por ser de código abierto resultó ser las que cuentan con SO Android y ofrece a los usuarios no sólo las ventajas actuales de la comunicación Web, sino que además amplía sus beneficios con el uso de las funcionalidades que ofrecen éstos dispositivos “*smartphones*” tales como: movilidad, captura y envío de imágenes, entre otras.

**Tabla 2. Aspectos funcionales de los Sistemas Operativos**

| Sistema Operativo                 | Apple iOS | Android     | Windows Phone        | Blackberry OS        | Symbian   |
|-----------------------------------|-----------|-------------|----------------------|----------------------|-----------|
| <b>Año de Lanzamiento</b>         | 2007      | 2008        | 2010                 | 2003                 | 1997      |
| <b>Soporte Flash</b>              | No        | Si          | No                   | Si                   | Si        |
| <b>Tienda de Aplicaciones</b>     | App Store | Google Play | Windows Market Place | Blackberry App World | OVI Store |
| <b>Aplicaciones</b>               | 650.000   | 450.000     | 50.000               | 30.000               | 50.000    |
| <b>Soporte de Memoria Externa</b> | No        | Si          | No                   | Si                   | Si        |
| <b>Fabricante Único</b>           | Si        | No          | No                   | Si                   | No        |
| <b>Variedad de Dispositivos</b>   | Baja      | Muy Alta    | Baja                 | Baja                 | Baja      |
| <b>Duración de Batería</b>        | Regular   | Regular     | Buena                | Buena                | Buena     |

Es por ello, que una vez realizadas las diferentes investigaciones y análisis mencionados anteriormente, se acordó, la realización de dos aplicaciones, una aplicación móvil nativa desarrollada en el sistema operativo Android, ya que éste demostró ser uno de los más utilizados con mayor demanda en la actualidad, por ser de código abierto. Ésta aplicación nativa va dirigida únicamente al usuario “Médico Rural”, dado que éste deberá utilizar la aplicación en lugares remotos por lo que necesitará una conexión “stand alone”, ventaja que ofrece por ser nativa.

Por otro lado el desarrollo de una aplicación Web móvil que permita su uso desde cualquier dispositivo móvil “*smartphone*” a través de un navegador Web, ya que permite a cualquier usuario utilizarla sin importar el sistema operativo que posea en su dispositivo.

El proyecto consistió en el desarrollo de una solución móvil comprendida por dos aplicaciones de software. La primera, una aplicación Web que permite a los usuarios a través de un navegador conectarse con el sistema de manera remota mediante sus dispositivos móviles. La segunda, una aplicación nativa desarrollada para el sistema operativo móvil: Android, permitiéndole a los usuarios que poseen este sistema operativo la conexión al sistema con una aplicación propia del dispositivo.

El sistema móvil desarrollado, está basado en el Sistema de Referencias Médicas existente en la actualidad. Se utilizó la estructura y la lógica del mismo para que ambos se complementaran y reflejaran los mismos resultados dependiendo del dispositivo utilizado, adicionalmente para esta versión móvil se permite la creación de un nuevo caso de consulta emitido desde el propio centro asistencial por parte del médico rural.

La composición del sistema se adapta al sistema web existente y al manejo del flujo de información actual. El flujo global se basa en que el sistema proporciona la realización de diagnósticos médicos por parte de especialista, para las diferentes patologías que puedan presentar los pacientes que se encuentran en los centros asistenciales del país.

Para el diseño de la interfaz gráfica se implementaron los formatos utilizados en el Sistema de Referencias Médicas ya existente, basados en colores amarillo, gris y azul, con el logo del proyecto SOS Telemedicina para Venezuela y el logo de la Universidad Central de Venezuela.

Los lenguajes y las herramientas de programación utilizadas fueron: HTML5, JavaScript, jQuery Mobile, Ajax, PHP, MySQL, Java y XML.

#### **4.4. Etapas de desarrollo aplicación Web móvil**

En este apartado se describen el conjunto de procedimientos llevados a cabo durante cada etapa de desarrollo del software de la aplicación Web móvil.

##### **4.4.1. Etapa I. Planificación**

Durante esta etapa como se mencionó en el capítulo anterior, se llevó a cabo el proceso de análisis de requerimientos del sistema, a partir de las necesidades presentes en el programa SOS Telemedicina para Venezuela y del Sistema de Referencias Médicas ya existente.

Una vez realizadas las diferentes investigaciones y análisis mencionados anteriormente, se propuso al Centro de Informática Médica, la realización de una aplicación Web móvil que permita su uso desde cualquier dispositivo móvil inteligente “*Smartphone*” a través de un navegador Web accediendo a la base de datos del Sistema de Referencias Médicas actual.

Para esta aplicación los requerimientos específicos se basan en los tres tipos de usuarios: médico triaje, médico especialista y médico rural con acceso a la aplicación Web móvil independiente del sistema operativo de cada dispositivo que posean.

Al establecer las necesidades propias de la aplicación y las ventajas que ofrecen los dispositivos móviles se consolidó la información obtenida y se definieron los requerimientos funcionales y no funcionales, así como la arquitectura del sistema, expuestos en el Capítulo III Marco Metodológico.

##### **4.4.2. Etapa II. Diseño**

Para el análisis de la aplicación se utilizó el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema, además ofrece un estándar para describir aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

A través de éste lenguaje se desarrollaron los diagramas del sistema tal como lo son: diagramas de casos de uso, diagrama de actividades, máquina de estados de la aplicación, diagramas de

clases y el modelo entidad – relación, mediante los cuales se especifica el funcionamiento del sistema y la interacción de los diferentes usuarios con el mismo.

- **Diagrama de casos de uso**

Mediante el diagrama de casos de uso representamos la forma en cómo el usuario (Actor) opera con la aplicación web móvil en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso) entre sí.

Los casos de uso se presentan mediante diferentes niveles que representan el nivel de detalle requerido. Nivel 0 (figura 17), Nivel 1 (figura 18) y Nivel 2 (ver figura A.1, Apéndice A)

El diagrama de casos de uso consta de los siguientes elementos:

- **Actores:** Los diferentes actores que interactúan con la aplicación son: médico triaje, médico especialista y médico rural. Ver tabla 3.

**Tabla 3. Actores identificados**

| Actor                      | Descripción   |
|----------------------------|---|
| <b>Médico Triage</b>       | Personal médico ubicado en la unidad de Triage capaz de asignar ó reasignar casos y emitir opiniones médicas. |
| <b>Médico Especialista</b> | Personal médico capaz de emitir opiniones médicas a las patologías presentadas.                               |
| <b>Médico Rural</b>        | Personal médico ubicado en los centros asistenciales remotos capaz de generar nuevos casos.                   |

- **Casos de Uso:** Las actividades que estos usuarios ejecutan en el sistema. (Ingresar al sistema, ver casos, tramitar casos, ver mis casos, ver respuestas, etc.).
- **Relaciones de Uso, Herencia y Comunicación:** Las tres relaciones principales entre los casos de uso son soportadas por el estándar UML, el cual describe notación gráfica para esas relaciones. Inclusión (*include*), Extensión (*extend*) y generalización.

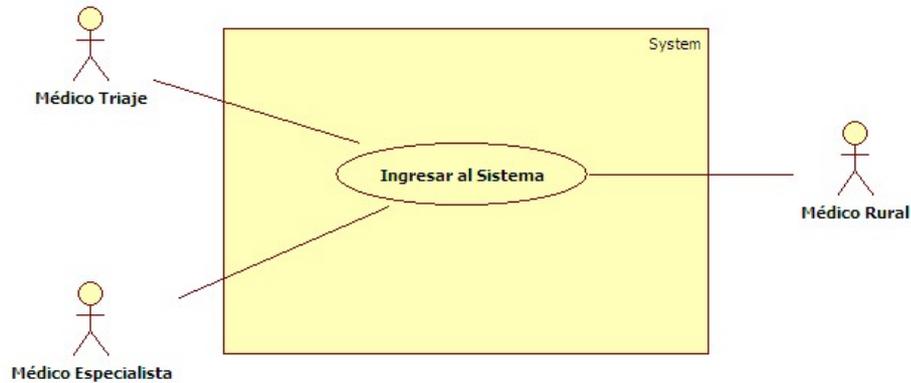


Figura 17. Diagrama de casos de uso nivel 0

En la tabla 4 se describen los casos de uso y sus descripciones correspondientes para el nivel 0 en cuanto al detalle de la aplicación web.

**Tabla 4. Casos de uso Nivel 0**

| Casos de Uso Nivel 0             |   |
|----------------------------------|---|
| <b>Pre-condición</b>             | Ninguna.  |
| <b>Nombre</b>                    | Ingresar al Sistema.  |
| <b>Actores</b>                   | Todos los usuarios (Médico Triage, Médico Especialista y Médico Rural)  |
| <b>Descripción</b>               | El usuario debe ingresar los datos de acceso al sistema para poder ser validados y así realizar las diferentes funcionalidades según su perfil. |
| <b>Casos de uso relacionados</b> | Ninguno   |
| <b>Entrada</b>                   | Datos del usuario   |
| <b>Salida</b>                    | Ingreso correcto al sistema   |

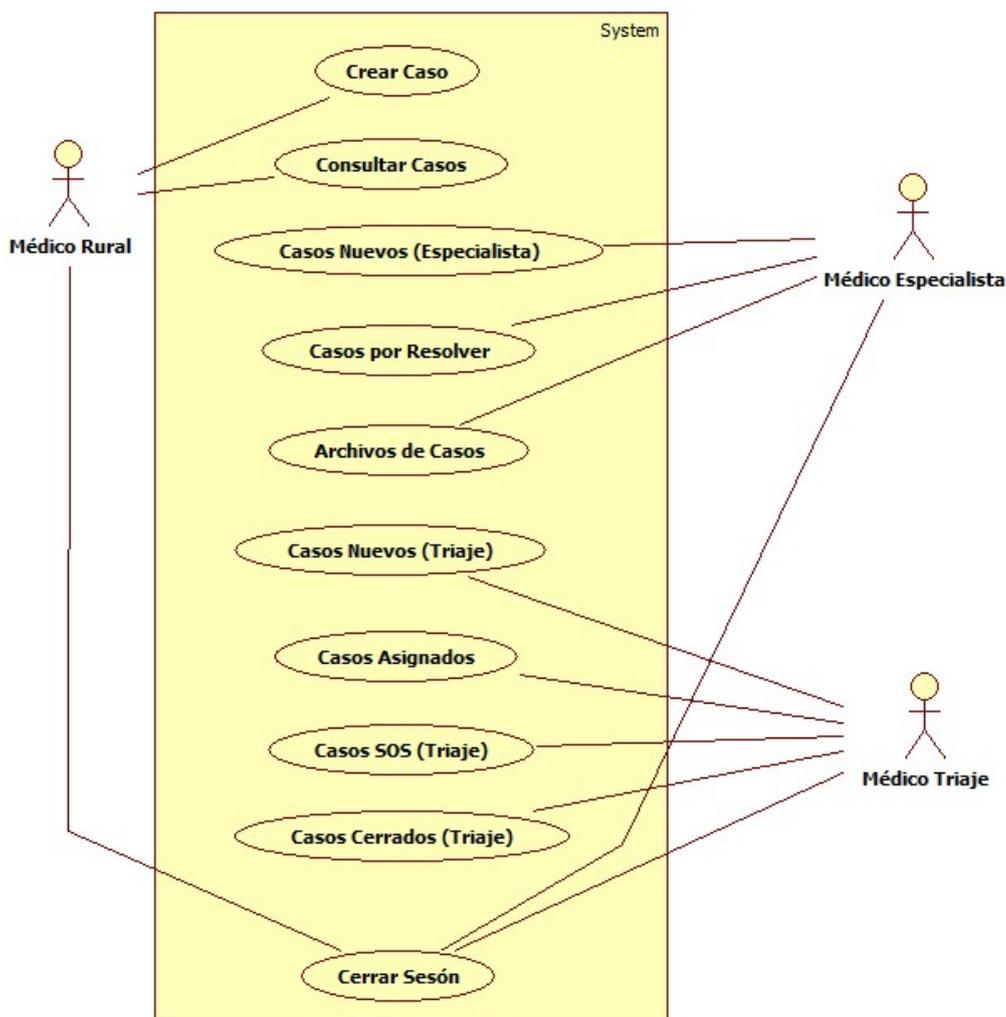


Figura 18. Diagrama de casos de uso Nivel 1

En la tabla 5 se detallan los casos de uso y casos asociados, correspondientes al nivel 1 del diseño de la aplicación web.

**Tabla 5. Casos de uso Nivel 1**

| Casos de Uso Nivel 1             |   |
|----------------------------------|---|
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.   |
| <b>Nombre</b>                    | Crear Caso.   |
| <b>Actores</b>                   | Médico Rural.   |
| <b>Descripción</b>               | El usuario puede crear un caso nuevo dentro del sistema.  |
| <b>Casos de uso relacionados</b> | Ninguno   |
| <b>Entrada</b>                   | Datos del nuevo caso a crear, descripción, archivos adjuntos y selección de centro asistencial. |
| <b>Salida</b>                    | Caso nuevo creado y registrado en el sistema.   |

Tabla 5. Casos de uso Nivel 1 (Continuación)

| Casos de Uso Nivel 1             |  |
|----------------------------------|--|
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Consultar Casos.   |
| <b>Actores</b>                   | Médico Rural.  |
| <b>Descripción</b>               | El usuario puede seleccionar los casos para consultarlos según sus estatus en los diferentes grupos (casos abiertos, casos SOS, casos cerrados)  |
| <b>Casos de uso relacionados</b> | Ver Casos Abiertos, Ver Casos SOS, Ver Casos Cerrados.   |
| <b>Entrada</b>                   | Ninguna.   |
| <b>Salida</b>                    | Visualización del detalle del caso seleccionado.   |
| <b>Casos de Uso Nivel 1</b>      |  |
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Consultar Casos Nuevos (Especialista).   |
| <b>Actores</b>                   | Médico Especialista.   |
| <b>Descripción</b>               | El usuario puede visualizar todos los casos nuevos que ha recibido y realizar las diferentes opciones permitidas en ellos (aceptar, rechazar, emitir opinión, ver opiniones y ver archivos). |
| <b>Casos de uso relacionados</b> | Aceptar Caso, Rechazar Caso, Emitir Opinión, Ver Opiniones y Ver Archivos.   |
| <b>Entrada</b>                   | Dependiendo de la operación a realizar.  |
| <b>Salida</b>                    | Cambio de estatus del caso.  |
| <b>Casos de Uso Nivel 1</b>      |  |
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Consultar Casos por Resolver.  |
| <b>Actores</b>                   | Médico Especialista.   |
| <b>Descripción</b>               | El usuario puede visualizar todos los casos asignados y que aún no se les ha emitido una opinión.  |
| <b>Casos de uso relacionados</b> | Emitir Opinión, Ver Opinión y Ver Archivos.  |
| <b>Entrada</b>                   | Ninguna  |
| <b>Salida</b>                    | Detalle del (los) casos por resolver.  |

Tabla 5.1. Casos de uso Nivel 1 (Continuación)

| Casos de Uso Nivel 1             |  |
|----------------------------------|--|
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Ver Archivo de Casos.  |
| <b>Actores</b>                   | Médico Especialista.   |
| <b>Descripción</b>               | El usuario puede consultar los casos asignados a su usuario.   |
| <b>Casos de uso relacionados</b> | Ninguno  |
| <b>Entrada</b>                   | Datos del nuevo caso a crear, descripción, archivos adjuntos y selección de centro asistencial.  |
| <b>Salida</b>                    | Caso nuevo creado y registrado en el sistema.  |
| <b>Casos de Uso Nivel 1</b>      |  |
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Consultar Casos Nuevos (Triage).   |
| <b>Actores</b>                   | Médico Triage.   |
| <b>Descripción</b>               | El usuario puede visualizar todos los casos nuevos que ha recibido y realizar las diferentes opciones permitidas en ellos (asignar, emitir opinión, ver opiniones y ver archivos). |
| <b>Casos de uso relacionados</b> | Asignar Caso, Emitir Opinión, Ver Opiniones y Ver Archivos.  |
| <b>Entrada</b>                   | Dependiendo de la operación a realizar.  |
| <b>Salida</b>                    | Cambio de estatus del caso.  |
| <b>Casos de Uso Nivel 1</b>      |  |
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Consultar Casos Asignados.   |
| <b>Actores</b>                   | Médico Triage.   |
| <b>Descripción</b>               | El usuario puede seleccionar los casos nuevos que se encuentran asignados y consultar su estatus.  |
| <b>Casos de uso relacionados</b> | Ver Casos Pendientes, Ver Casos Rechazos y Ver Casos Aceptados.  |
| <b>Entrada</b>                   | Ninguna.   |
| <b>Salida</b>                    | Visualización del detalle del caso seleccionado.   |

Tabla 5.2. Casos de uso Nivel 1 (Continuación)

| Casos de Uso Nivel 1             |  |
|----------------------------------|--|
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Consultar Casos SOS.   |
| <b>Actores</b>                   | Médico Triage.   |
| <b>Descripción</b>               | El usuario puede seleccionar los casos provenientes de SOS-HME y consultar su estatus. |
| <b>Casos de uso relacionados</b> | Ver Opiniones y Ver Archivos.  |
| <b>Entrada</b>                   | Ninguna.   |
| <b>Salida</b>                    | Visualización del detalle del caso seleccionado.                                       |
| <b>Casos de Uso Nivel 1</b>      |  |
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Consultar Casos Cerrados.  |
| <b>Actores</b>                   | Médico Triage.   |
| <b>Descripción</b>               | El usuario puede seleccionar los casos que se encuentran cerrados.                     |
| <b>Casos de uso relacionados</b> | Ver Opiniones y Ver Archivos.  |
| <b>Entrada</b>                   | Ninguna.   |
| <b>Salida</b>                    | Visualización del detalle del caso seleccionado.                                       |
| <b>Casos de Uso Nivel 1</b>      |  |
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Salir del Sistema.   |
| <b>Actores</b>                   | Todos los usuarios (Médico Triage, Médico Especialista y Médico Rural)                 |
| <b>Descripción</b>               | Permite al usuario salir del sistema de manera exitosa.                                |
| <b>Casos de uso relacionados</b> | Ninguno  |
| <b>Entrada</b>                   | Ninguna.   |
| <b>Salida</b>                    | Sesión de usuario cerrada.   |

- **Diagrama de Clases**

Los diagramas de clases muestran las diferentes clases que componen la Aplicación Web Móvil y cómo se relacionan unas con otras. Muestra las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases «conocen» a qué otras clases o qué clases «son parte» de otras clases. Ver figura 19.

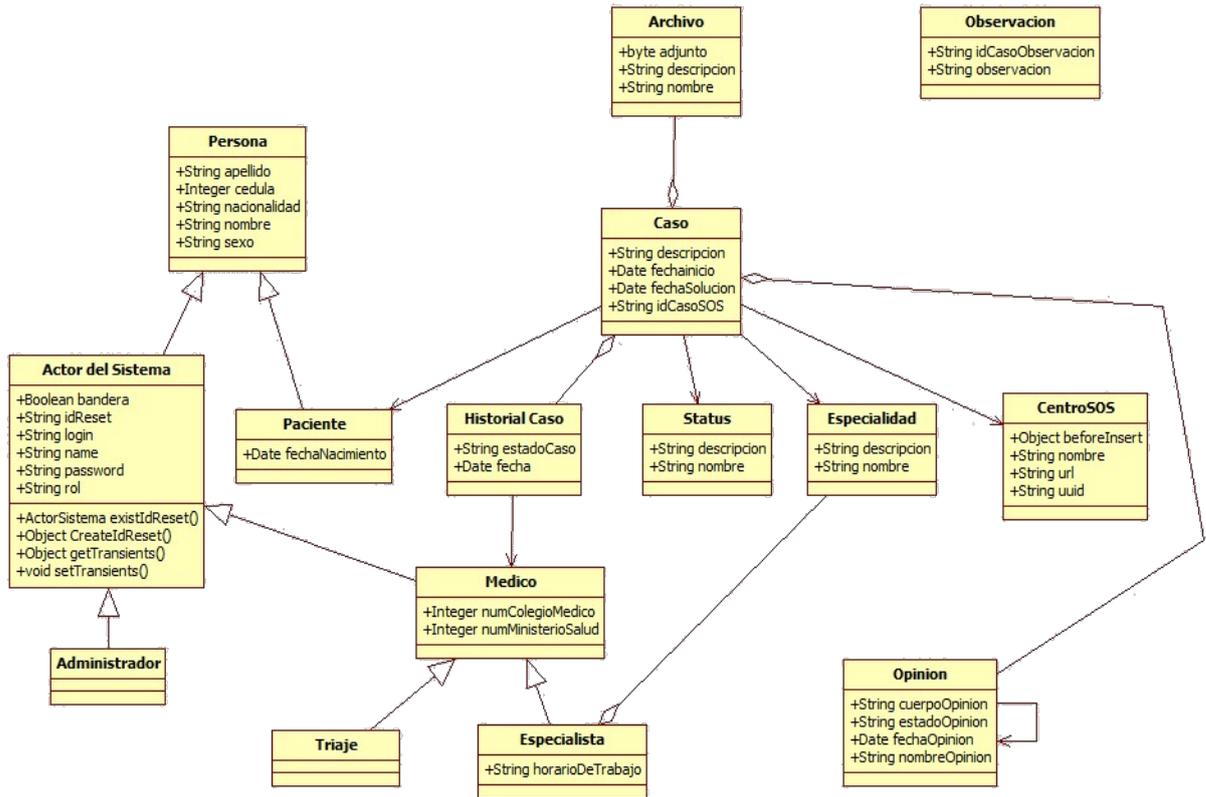


Figura 19. Diagrama de clases del Sistema Móvil de Referencias Médicas

- **Diagrama de Actividad**

Este diagrama muestra el comportamiento desde una actividad hacia otra actividad dentro del sistema móvil, es decir, muestra el flujo de control general. El diagrama de actividad presenta un único punto de inicio y un único punto de término, aunque puede tener más, siempre que cumpla con la lógica requerida dentro del sistema. Ver figura 20. Para el resto de los diagramas ver apéndice A, desde Figura A.2 – Figura A.9.

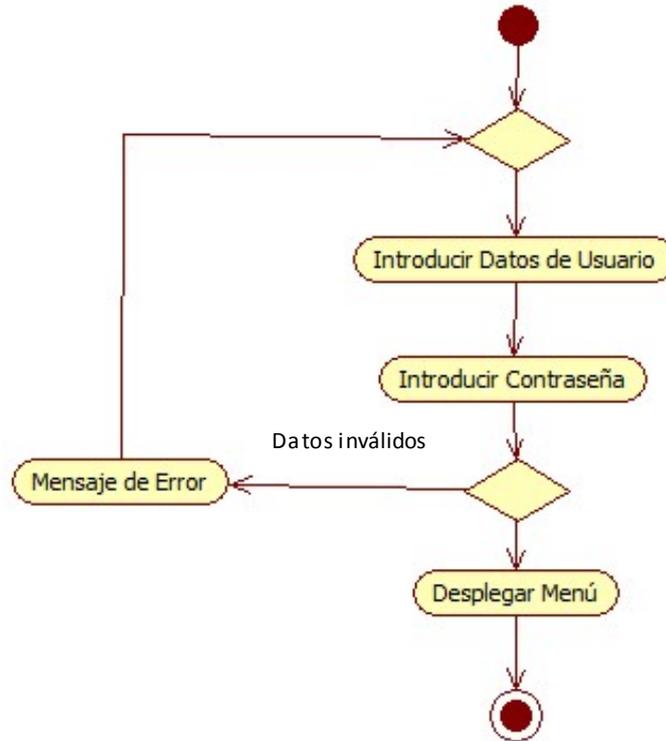


Figura 20. Diagrama de actividad para el caso de uso “Ingresar al Sistema”

- **Máquina de Estados**

Las máquinas de estados representa un conjunto de estados que atraviesa un objeto durante su tiempo de vida dentro del Aplicación Web Móvil. La máquina de estados, define el proceso y cada uno de los estados por los cuales pasaran las referencias médicas; ésta contempla el proceso completo de las mismas, iniciando por el estado que toman al ser referenciadas y finalizando por el ultimo estado que toman una vez éstas se encuentran resueltas. Ver figura 21.

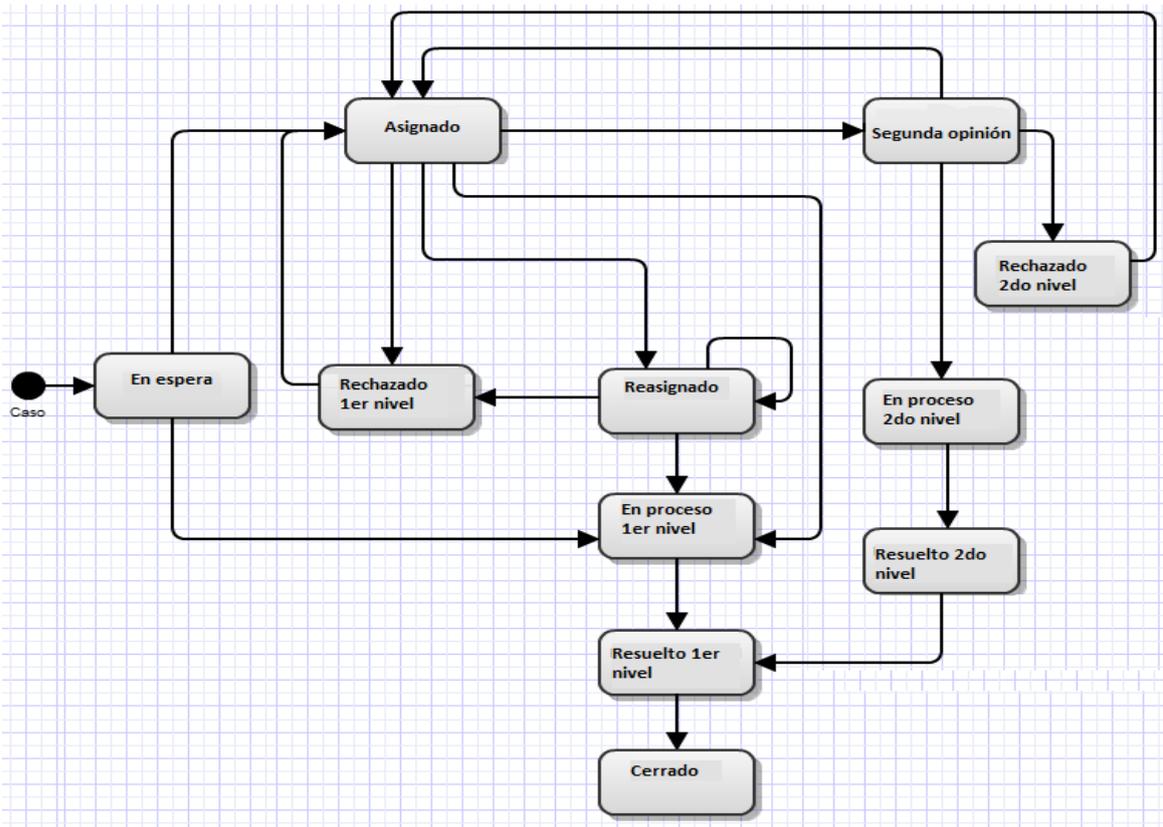


Figura 21. Máquina de Estados del Sistema de Referencias Médicas

- **Modelo Entidad – Relación**

Este diagrama permite a través del modelado de datos representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades. Se diseñó el siguiente modelo para crear a partir de él, una base de datos que contenga todas las tablas y atributos necesarios para almacenar los datos que proporciona el módulo de Referencia Médicas y lograr con ello el manejo de manera óptima de los mismos. Ver Apéndice A, Figura A.10.

**Diseño de la interfaz gráfica**

La interfaz original del fue totalmente rediseñada para adaptarla a las funcionalidades de los dispositivos móviles. Fue necesario cambiar la interfaz gráfica original del por dos motivos principales; mejorar la experiencia del usuario y adaptar las vistas al concepto visual de SOS Telemedicina dentro del ambiente móvil.

En una mesa de trabajo se definieron los requerimientos de las vistas, dando como resultado una propuesta de interfaz presentada. La gama de colores varía entre el amarillo, blanco y azul, además de plasmar los logos de SOS Telemedicina para Venezuela y el de la Universidad Central de Venezuela

La interfaz gráfica se desarrolló utilizando las bondades del Framework jQuery Mobile que permitió diseñar entre otras cosas:

- Botones con formato enriquecido.
- Cuadros de diálogo.
- Validación de Campos de Formularios
- Interfaz con diseño Responsivo.
- Transiciones animadas entre pestañas.
- Adaptabilidad de la interfaz al girar la pantalla Horizontal y Vertical
- Lista de elementos.
- Lista desplegable de elementos.
- Carga de Archivos e Imágenes.

A continuación se muestran las principales vistas del sistema, el menú de navegación de las vistas está compuesto por las opciones principales según cada usuario. Desde cualquiera de la vistas el usuario puede cerrar la sesión y volver a la pantalla de inicio.

## Vista sesión del usuario Médico Triaje

La interfaz “Médico de Triaje”, ver figura 22, muestra la pantalla principal del usuario luego de su inicio de sesión, y mediante una lista presenta todas las funcionales que tiene permitidas según su perfil dentro del sistema.



Figura 22. Interfaz “Médico Triaje”

## Vista sesión de usuario Médico Especialista

La figura 23, muestra la interfaz del usuario “Médico Especialista”, donde se despliegan mediante una lista todas las funcionalidades permitidas para dicho usuario dentro del sistema Web.



Figura 23. Menú Principal Médico Triaje

## Vista sesión de usuario Médico Rural

La figura 24, muestra la interfaz del usuario “Médico Rural”, donde muestra la lista desplegable que contiene todas las funcionalidades permitidas para dicho usuario dentro de la Aplicación Web Móvil.



Figura 24. Menú Principal Médico Rural

### 4.4.3. Etapa III. Codificación

En el proceso de codificación se implementó la arquitectura propia del sistema y se desarrollaron los módulos y funcionalidades de cada usuario, a través de diferentes herramientas de software.

#### 4.4.3.1. Arquitectura

La arquitectura del sistema desarrollado está compuesta por:

- Base de datos del Sistema de Referencias Médicas (Base de datos MySQL), la cual contiene toda la información almacenada de registros clínicos, usuarios, etc.
- Servicios Web, que permiten la conexión a la base de datos desde el dispositivo móvil.
- Dispositivo móvil, mediante el cual el usuario se conecta a la aplicación a través de Internet.

A continuación en la figura 25 se describe la arquitectura definida para la aplicación Web móvil.

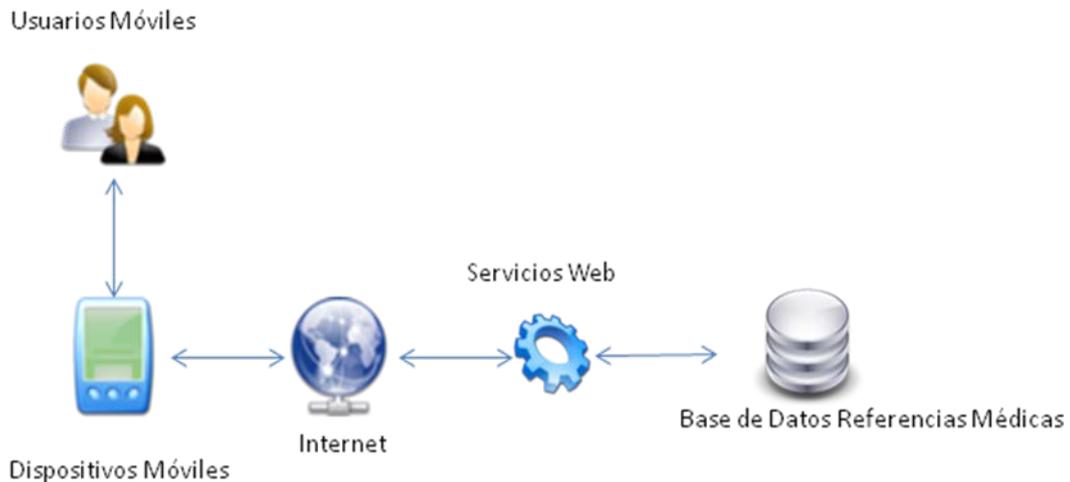


Figura 25. Arquitectura aplicación Web Móvil

Para el desarrollo de la aplicación Web Móvil y su codificación, se utilizaron las siguientes herramientas de software:

#### 4.4.3.2. HTML5 (HyperText Markup Language, versión 5)

HTML es un estándar que, en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página Web, como texto, imágenes, entre otros. [18]

El lenguaje HTML basa su filosofía de desarrollo en la referenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, etc.), este no se incrusta directamente en el código de

la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página Web contiene sólo texto mientras que recae en el navegador Web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página Web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador Web actualizado.

HTML5 es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. Establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios Web modernos. Algunos de ellos son técnicamente similares, pero tienen un significado semántico distinto. Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos <audio> y <video>.

#### **4.4.3.3. JavaScript**

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo y dinámico. [19]

Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador Web permitiendo mejoras en la interfaz de usuario y páginas Web dinámicas aunque existe una forma de JavaScript del lado del servidor. Su uso en aplicaciones externas a la Web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

El uso más común de JavaScript es escribir funciones embebidas o incluidas en páginas HTML y que interactúan con el Modelo de Objetos del Documento (DOM , *Document Object Model*) de la página.

Dado que el código JavaScript puede ejecutarse localmente en el navegador del usuario (en lugar de en un servidor remoto), el navegador puede responder a las acciones del usuario con rapidez, haciendo una aplicación más sensible. Por otra parte, el código JavaScript puede detectar acciones de los usuarios que HTML por sí sola no puede, como pulsaciones de teclado. La tendencia cada vez mayor por el uso de la programación Ajax explota de manera similar esta técnica. [19]

#### **4.4.3.4. AJAX (Asynchronous JavaScript And XML )**

Es una técnica de desarrollo Web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.[20]

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (*scripting language*) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XML HttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML. [20]

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Modelo de Objetos del Documento (DOM , *Document Object Model*).

#### **4.4.3.5. jQuery**

jQuery es un Framework de JavaScript para facilitar, entre otros, el acceso a los elementos del DOM, permite interactuar con los documentos HTML, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas Web. [21]

La característica principal de la biblioteca es que permite cambiar el contenido de una página Web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX.

#### **4.4.3.6. jQuery Mobile**

Desarrollado por el proyecto jQuery, es un Framework para desarrollo Web está optimizado para ser usado en dispositivos inteligentes “smartphones”. Es una librería para JavaScript y HTML5 por lo que puede usarse para crear aplicaciones para casi cualquier navegador Web reciente, sin embargo, está pensado específicamente para ser usado en dispositivos móviles. [22]

jQuery Mobile permite generar aplicaciones cuya apariencia será siempre la misma independientemente del dispositivo desde el que acceda un usuario siempre que este usuario acceda desde un dispositivo que acepte HTML5.

#### **4.4.3.7. PHP (Hypertext Pre-processor)**

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo Web de contenido dinámico. El código es interpretado por un servidor Web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores Web al igual que en casi todos los sistemas operativos y plataformas. [23]

#### **4.4.3.8. Sistema Manejador de Base de Datos MySQL**

Es un sistema de gestión de bases de datos relacionales, Es un manejador ideal para aplicaciones Web, pues garantiza una rápida lectura sobre los datos, y tiene un mejor desempeño en un entorno de baja concurrencia de escritura. [24]

MySQL es un servidor multi-usuarios muy rápido y robusto de ejecución de instrucciones en paralelo, es decir, que múltiples usuarios distribuidos a lo largo de una red local o Internet podrá ejecutar distintas tareas sobre las bases de datos localizadas en un mismo servidor. Utiliza el lenguaje SQL (*Structured Query Language*) que es el estándar de consulta a bases de datos a nivel mundial.

Según las cifras del propio fabricante, existen más de seis millones de copias de MySQL funcionando en la actualidad, lo que supera la base instalada de cualquier otra herramienta de bases de datos.

#### **4.4.3.9. WampServer**

WAMP acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas [25]:

- Windows, como sistema operativo.
- Apache, como Servidor web.
- MySQL, como gestor de bases de datos.
- PHP (generalmente), Perl, o Python, como lenguaje de programación.

El uso de un WAMP permite servir páginas html a internet, además de poder gestionar datos en ellas, al mismo tiempo un WAMP, proporciona lenguajes de programación para desarrollar aplicaciones web.

WampServer es un entorno de desarrollo web para Windows con el que se puede crear aplicaciones web con Apache, PHP y bases de datos MySQL.

### **Características**

Provee a los desarrolladores con los cuatro elementos necesarios para un servidor web: un Sistema Operativo (Window), un manejador de base de datos (MySQL), un software para servidor web (Apache) y un software de programación script Web (PHP (generalmente), Python o PERL), debiendo su nombre a dichas herramientas. WampServer posee licencia OpenSource o código abierto por lo que se puede descargar de forma gratuita.

- **Configuración del entorno de la aplicación**

A continuación se describe el proceso de instalación y configuración de los componentes de software y hardware necesarios, para garantizar la correcta ejecución del Sistema Móvil de Referencias Médicas específicamente desde la aplicación web mobile.

### **WampServer**

#### **Configuración del servidor**

Motivado a que el Sistema de Referencias Médicas versión Web móvil es una aplicación Web, es necesario que instalemos un servidor Web para poder alojar los archivos de nuestro sistema y así poder ejecutar la aplicación.

A continuación se describe el proceso de configuración e instalación bajo la plataforma Microsoft Windows, de las herramientas necesarias para la configuración del equipo servidor:

- Configuración de Servidor web (En nuestro caso WampServer).
- Configuración de la conexión a la base de datos MySQL del Sistema de Referencias Médicas.

### Configuración del servidor Web

Al instalar el WampServer procedemos a colocar la carpeta de nuestra aplicación llamada SegundaOpinion en la carpeta en donde se instaló el servidor apache, específicamente en la carpeta www que sería nuestra carpeta pública desde donde accederemos a la aplicación. La ruta sería la siguiente C:\wamp\www\SegundaOpinion.

Configuración de la conexión a la base de datos MySQL del Sistema de Referencias Médicas para PC. Para conectarnos con la base de datos del Sistema SOS-TRIAJE, debemos acceder a la clase Conector del archivo "class.php" ubicada dentro del directorio de nuestro sistema C:\wamp\www\SegundaOpinion\Modelo.

La clase Conector es la que permitirá realizar la autenticación y conexión con la base de datos del sistema para poder realizar todas las operaciones con los datos requeridos en nuestra aplicación.

Una vez en la clase conector debemos registrar los parámetros de la conexión con la base de datos MySql como lo son:

- El Host, el cual es la dirección ip en donde está almacenada la Base de Datos.
- El Usuario y Password que permite el acceso seguro a la base de datos.
- El nombre de la Base de Datos a seleccionar.

Para el sistema los parámetros de configuración fueron los siguientes (figura 26):

```
<?php  
  
class Conector{  
var $host = "localhost";  
var $user = "root";  
var $pass = "";  
var $bd = "sos_triaje";  
var $conec;
```

Figura 26. Configuración de parámetros

Fuente: <http://www.wampserver.com>

## **Configuración del Cliente**

Al ser una aplicación web, los requisitos del lado del cliente para poder correr la aplicación serían:

### **Navegador Web**

- Habilitar Soporte para Javascript (Preferiblemente mas no es limitante)
- Conexión de datos (Wi-fi, plan de datos)

#### **4.4.3.10. Módulos y funcionalidades**

La aplicación Web móvil está compuesta por un conjunto de tres módulos: Médico Triage, Médico Especialista y Médico Rural, cada uno de ellos con un conjunto de funcionalidades que le permiten a cada usuario interactuar con el sistema.

### **Funcionalidades Generales**

Existen un grupo de funcionalidades que son comunes para todos los módulos del sistema, con características específicas según el tipo de usuario. Las funcionalidades generales son:

- **Ingresar al Sistema**

Permite al usuario que se está autenticando acceder al sistema, según su perfil. El desarrollo de ésta funcionalidad es el siguiente:

#### **Cliente**

Desde el punto de vista de la interfaz de usuario se diseñó la interfaz de ingreso al sistema de forma tal que sea usable e intuitiva para el usuario final, implementada en ella dos campos: “Usuario” y “Contraseña” además de un botón “Ingresar”.

En los campos de texto se aprovechó las bondades que brinda JQueryMobile para la validación de campos de formulario, pudiendo entonces validar que los campos “Usuario” y “Contraseña” no estén vacíos. Ver figura 27.



Figura 27. Pantalla de Inicio

Utilizando el atributo “placeholder” de HTML5 se colocó una pequeña introducción en cada campo de texto para que el usuario sepa cuál valor se le está solicitando.

Cabe destacar que todas estas validaciones se realizan del lado del usuario (cliente) lo que permite que la petición que se realizará hacia el servidor ya venga verificada y validada y así establecer parámetros de seguridad para evitar posibles ataques como inyección de código SQL en formularios por ejemplo.

Una vez que se validan los datos del lado del usuario, se realiza la petición vía protocolo HTTP hacia el servidor, presionando el botón “Ingresar”. Esta petición se envía hacia el servidor y la valida un archivo .PHP llamado “auth.php” que es quién manejará dicha solicitud.

### **Servidor**

El archivo “auth.php” toma los campos enviados por el usuario por medio de las variables \$\_POST que es la manera como fueron enviados los datos en la petición del usuario y obtiene la información que deberá ser validada.

El archivo “auth.php” invoca a la clase “class.php” quien es la que manejará todo lo referente a la conexión a la base de datos y las operaciones que se pueden realizar sobre la misma como lo son “Conexión” “Ejecutar Consulta” “Cerrar Conexión” entre otras.

El archivo “auth.php” toma los datos de la variable \$\_POST y realiza la consulta contra la base de datos en el servidor y valida la coincidencia de los mismos con los almacenados en la base de datos sos\_triaje.

Si los datos no coinciden, se enviará un mensaje de error de fallo de autenticación indicándole al usuario que el usuario y contraseña son incorrectos y que deben ser verificados.

Si los datos coinciden, es decir si la autenticación es exitosa, se crean unas variables de sesión llamadas \$\_SESSION en donde se conservan los datos del usuario conectado durante todas las pestañas durante su conexión con el sistema.

Una vez creada la sesión del usuario, se valida el rol que tiene dicho usuario en el sistema (Médico rural, Médico especialista y Médico de Triage). Dependiendo del tipo de usuario, se desplegará el menú correspondiente con las opciones posibles por perfil seleccionado y se devuelve el control a la interfaz de usuario en el menú mencionado.

- **Ver Opiniones**

Al seleccionar un caso para ver los detalles que lo describen, se encuentra un campo llamado “Opiniones”. Este campo seleccionable nos permite observar las opiniones asociadas al caso en cuestión. Para ello se implementó una interfaz con la información del caso a solicitar, y en la parte inferior, unos botones que permiten ir seleccionando las opiniones asociadas al caso. Al seleccionar dicha opinión, se realiza una petición a la base de datos del sistema, específicamente a la tabla opiniones y devuelve el detalle de la opinión solicitada a la interfaz. Ver figura 28.

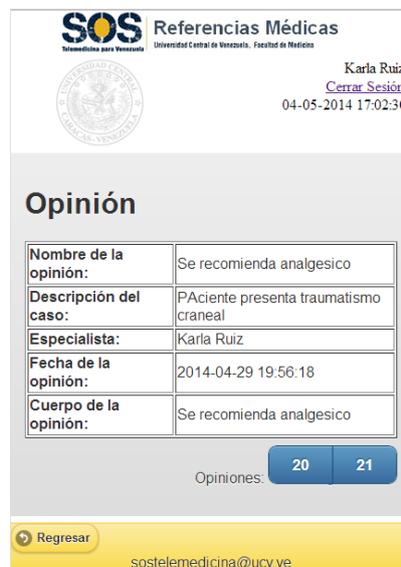


Figura 28. Detalle de Opinión

- **Ver Archivos**

Al seleccionar un caso para ver los detalles que lo describen, se encuentra un campo llamado “Archivos”. Este campo permite tener acceso a los archivos almacenados en la base de datos. Al seleccionar dicho archivo, se realiza una petición a la base de datos del sistema, específicamente a la tabla archivos que retorna bien sea la imagen que se tomó asociada al caso, así como pudieran ser los exámenes de RX o algún otro archivo de interés

- **Cerrar Sesión**

Salir de forma controlada del sistema cerrando la sesión del usuario actual.

Se puede salir del sistema bien sea presionando el botón “Cerrar Sesión” desde el menú o seleccionando la opción “Cerrar Sesión” en la parte superior derecha de la pantalla.

En ambos casos se envía una petición al servidor y se invoca al archivo “salir.php” quien valida primero las credenciales del usuario autenticado. Paso seguido elimina las variables de sesión y cualquier dato que se había almacenado en la sesión del usuario. Luego de esto cierra cualquier conexión remanente con la base de datos invocando a la clase “class.php” y finalmente redireccione a la pantalla principal para una nueva autenticación.

### **Módulo Médico Triage**

Este módulo esta desarrollado para el usuario perteneciente a la unidad de Triage, el cual funciona como intermediario en la comunicación entre el médico rural y el médico especialista, asignando a este último los casos generados dentro del sistema ó resolviéndolos. Además de las funcionalidades generales que posee el usuario de Triage, también se encuentran:

- **Casos Nuevos**

Permite consultar los casos que están en Triage en estatus “En espera”, es decir que están pendientes por ser asignados. Cada caso muestra a su vez detalles como su descripción, estado actual del caso, Tiempo que tiene en espera, Datos del médico que creó el caso y del Paciente que tiene asociado dicho caso, opiniones y archivos que contenga el mencionado caso.

## **Cliente**

En la interfaz “Casos Nuevos” se implementó una lista desplegable en JQueryMobile que nos permite ver un resumen del contenido y cuando la seleccionamos se expande mostrando todo el detalle a continuación. Esto permite incluir gran cantidad de ítems en forma de lista desplegable o acordeón en una misma pantalla ideal para dispositivos móviles sin necesidad de interactuar con diferentes pantallas para obtener la información.

## **Servidor**

La interfaz del usuario desplegada “Casos Nuevos” contiene las variables de sesión del Médico autenticado actual. La interfaz se comunica en Backend con el controlador “usuarioCasos.php” a quien le manda las variables de sesión de dicho médico, y a su vez se comunica por medio de la clase “class.php” con la base de datos para realizar la consulta por medio de la sentencia SQL y así encontrar todos los casos que tiene asociado dicho médico especialista.

Esta consulta retorna una serie de datos que son recorridos en el código “PHP” y son mostrados en la interfaz nuevamente devolviéndole así el control a la misma.

En la interfaz gráfica que nos muestra el detalle del caso se tiene la posibilidad de acceder a diferentes funcionalidades las cuales son:

- “Asignar Caso”: el cual permite ir a la interfaz de Asignación / Reasignación del caso para que el mismo sea asociado a algún médico Especialista (Esta funcionalidad será explicada más adelante).
- “Emitir Opinión”: el cual es un formulario con el botón “Enviar” el cual permite dar respuesta al caso que estamos viendo en ese momento y cuya implementación se explicará más adelante.
- Ver Opiniones
- Ver Archivos

La interfaz gráfica de esta funcionalidad la pueden encontrar en la figura 29.



**SOS Referencias Médicas**  
Telemedicina para Venezuela



Universidad Central de Venezuela - Facultad de Medicina

Marcos Perez |  
[Cerrar Sesión](#)  
 10-05-2014 16:52:37

## Médico Triaje

Casos nuevos

- Caso: Paciente ingresa con arritmia cardiaca

|                       |  |
|-----------------------|--|
| Descripción del caso: | Paciente ingresa con arritmia cardiaca |
| Estado:               | En espera desde 2014-05-08 10:03:06    |
| Tiempo de Espera:     | 2 días                                 |
| Especialidad:         |  |
| Nombre Medico:        | Vanessa Rodriguez                      |
| Nombre Paciente:      | Paciente Generico                      |
| CI Paciente:          | 1234567                                |
| Opiniones:            |  |
| Archivos:             |  |

Cuerpo de la Opinión

Título de Opinion

Adjuntar:

Seleccionar archivo

Ningún archivo seleccionado

✓ Enviar

≡ Asignar Caso

Figura 29. Pantalla Casos Nuevos

- **Emitir Opinión**

Permite al usuario médico de triaje emitir una opinión sobre un caso en particular.

### Cliente

Desde el punto de vista de la interfaz de usuario se diseñó la pantalla de emisión de una opinión que le permita así resolver el caso. Para ello tenemos los siguientes campos: “Título de Opinión”, “Cuerpo de la opinión”, “Adjuntar” y un botón “Enviar” como se puede apreciar en la figura 28.

En el campo de texto “Cuerpo de la Opinión” aprovechamos las bondades que nos brinda JQueryMobile que permite que ese campo sea tan extenso como se necesite redimensionándose así al tamaño de la pantalla haciendo más cómodo el ingreso de datos desde cualquier dispositivo móvil.

Para el campo adjuntar se implementó la posibilidad de cargar un archivo desde cualquier directorio del sistema y adjuntarlo para ser enviado vía formulario gracias al atributo `enctype="multipart/form-data"` que nos permite enviar archivos vía protocolo http específicamente por el método POST.

### **Servidor**

Del lado del servidor se toma la petición del usuario en el archivo “ctrlOpinionTria.php” quien es el controlador y va a obtener todos los datos de la petición.

En este controlador “ctrlOpinionTria.php” se obtiene el archivo enviado y se almacena en la variable `$_FILES` la cual es una Variables de Carga de Archivos HTTP que nos permitirá manipular dicho archivo y codificarlo en nuestro caso en un arreglo de Bytes para almacenarlo en la base de datos en un campo tipo BLOB.

Una vez hecho esto se almacenan los datos enviados y se devuelve el control a la interfaz.

- **Asignar Casos / Reasignar Casos**

Consiste en asignar un caso que está en triaje a un especialista bien sea filtrando por su especialidad o por el nombre del especialista.

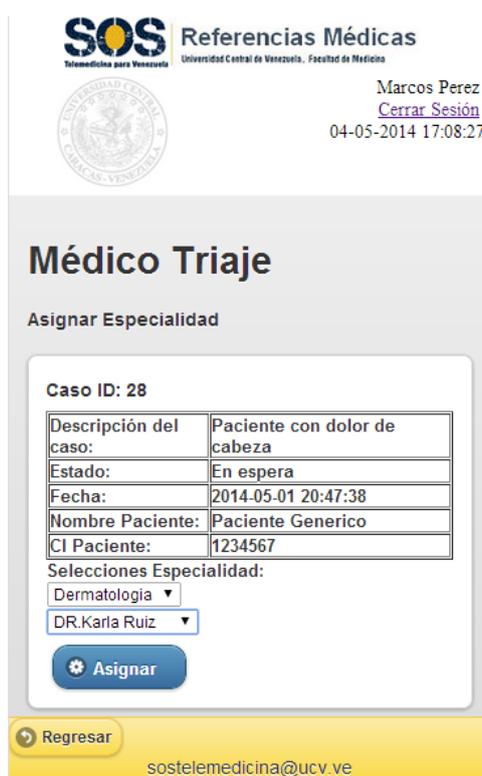
### **Cliente**

Desde el punto de vista de la interfaz de usuario se diseñó la pantalla de asignación o reasignación de un caso que le permita así al médico de triaje asignar el caso a un especialista para que el caso sea resuelto a la brevedad posible. Para ello tenemos los siguientes campos: “Especialidad”, “Especialista” y un botón “Asignar”.

En la lista seleccionable “Especialidad” obtenemos los datos desde la base de datos del sistema específicamente desde la tabla “Especialidad” la cual contiene todas las especialidades registradas en el sistema.

La lista seleccionable “Especialista” nos devuelve los nombres de los especialistas asociados a la especialidad seleccionable en la lista anterior.

Para ello aprovechamos las bondades de la tecnología AJAX que permite cargar el resultado de la selección de una primera lista desplegable en una segunda lista desplegable dependiente, y sin la necesidad de recargar toda la página, ya que la interfaz es actualizada solamente en el campo necesario. La interfaz gráfica de esta funcionalidad la pueden encontrar en la figura 30.



**SOS** Referencias Médicas  
Telemedicina para Venezuela Universidad Central de Venezuela, Facultad de Medicina

Marcos Perez |  
[Cerrar Sesión](#)  
04-05-2014 17:08:27

## Médico Triage

Asignar Especialidad

Caso ID: 28

|                       |                              |
|-----------------------|------------------------------|
| Descripción del caso: | Paciente con dolor de cabeza |
| Estado:               | En espera                    |
| Fecha:                | 2014-05-01 20:47:38          |
| Nombre Paciente:      | Paciente Generico            |
| CI Paciente:          | 1234567                      |

Selecciones Especialidad:

Dermatología ▾  
DR. Karla Ruiz ▾

sosteledicina@ucv.ve

Figura 30. Pantalla Asignar / Reasignar Caso

## Servidor

Del lado del servidor se toma la petición del usuario en el archivo “asignaEspecialidad.php” quien es el controlador y va a obtener todos los datos de la petición. Una vez obtenidos los datos, se almacenan y se devuelve el control a la interfaz.

Cabe destacar que como precondition, el usuario debe seleccionar una especialidad para poder asignar el caso a un especialista. En caso de no seleccionar alguna especialidad, el sistema la asignará a la especialidad por defecto que aparece en la lista, y seleccionará un especialista de esa especialidad de forma aleatoria.

Para el caso de la reasignación de un caso, el mismo debe cumplir la pre-condición que debe ser reasignado a otro médico de esa misma especialidad, por ejemplo un caso de Dermatología solo puede ser reasignado a otro médico de Dermatología.

- **Casos Asignados**

Permite consultar los casos que están en Triage que fueron asignados pero que no han sido resueltos por distintos motivos. Esta funcionalidad se divide en tres grandes rubros:

- Casos Pendientes: todos los casos que fueron asignados a un especialista, pero que aún no han sido aceptados por el médico y por ello se encuentra en estatus “Asignados”.
- Casos Rechazados: todos los casos que fueron asignados a un especialista, pero que fueron rechazados por este último, por lo que el caso se encuentra en estatus “Rechazado 1er nivel” o “Rechazado 2do nivel”.
- Casos Aceptados: todos los casos que fueron aceptados por un especialista, pero que aun no tienen respuesta y por ello se encuentran en estatus “En proceso 1er nivel” o “En proceso 2do nivel”

Cada caso muestra a su vez detalles como su descripción, estado actual del caso, Tiempo que tiene en espera, Especialidad a la que fue asignado, Especialista que está atendiendo el caso, Nombre Medico Atención Primaria que creó el caso y datos del Paciente que tiene asociado dicho caso, opiniones y archivos que contenga el mencionado caso.

En la interfaz gráfica que nos muestra el detalle del caso se tiene la posibilidad de acceder a diferentes funcionalidades las cuales son:

- “Reasignar Caso”: el cual permite ir a la interfaz de Asignación / Reasignación del caso para que el mismo sea asociado a algún médico Especialista.
- Para los casos en estatus “Asignados” o “En proceso” con más de 5 días, el sistema mostrará el botón de “Reasignar” que permite reasignar el caso a otro especialista de la misma especialidad a la que estaba asociado el caso original. (El valor de 5 días es parametrizable, pero actualmente está configurado este valor para efectos de nuestro trabajo especial de grado). En caso de que el tiempo sea inferior a 5 días, este botón no será mostrado.
- Para los casos en estatus Rechazados el botón de reasignar está siempre disponible, indistintamente el tiempo que lleve el caso en ese estatus.

- Ver Opiniones
- Ver Archivos

- **Casos SOS**

Permite consultar los casos que están en Triage que fueron resueltos por Especialistas, es decir que se encuentran en estatus “Resuelto 1er Nivel” y Resuelto 2do Nivel”. Esta funcionalidad permite tener un histórico de los casos resueltos que ya se encuentran en la base de datos.

Cada caso muestra a su vez detalles como su descripción, estado actual del caso, Tiempo que tiene en espera, Especialidad a la que fue asignado, Especialista que está atendiendo el caso, Nombre Medico Atención Primaria que creó el caso y datos del Paciente que tiene asociado dicho caso, opiniones y archivos que contenga el mencionado caso.

En la interfaz gráfica que nos muestra el detalle del caso se tiene la posibilidad de acceder a diferentes funcionalidades las cuales son:

- Ver Opiniones
- Ver Archivos

- **Casos Cerrados**

Permite consultar los casos que están en Triage que fueron resueltos por Especialistas y a su vez fueron Cerrados, es decir que se encuentran en estatus “Cerrados”. Esta funcionalidad permite tener un histórico de los casos resueltos que ya se encuentran en estatus cerrado la base de datos del sistema.

Cada caso muestra a su vez detalles como su descripción, estado actual del caso, Tiempo de duración total del caso, Especialidad a la que fue asignado, Especialista que resolvió el caso, Nombre Medico Atención Primaria que creó el caso y datos del Paciente que tiene asociado dicho caso, opiniones y archivos que contenga el mencionado caso.

En la interfaz gráfica que nos muestra el detalle del caso se tiene la posibilidad de acceder a diferentes funcionalidades las cuales son:

- Ver Opiniones
- Ver Archivos

## **Módulo Médico Especialista**

Este módulo está desarrollado para el usuario Médico Especialista, mediante el cual el usuario puede ejecutar las diferentes funcionalidades según su perfil dentro del sistema. Además de las funcionalidades generales que posee el usuario, también se encuentran:

- **Casos Nuevos**

Permite consultar los casos que fueron asignados al médico especialista y se encuentran en estatus “En espera”, es decir que están en espera de que el médico Especialista acepte o rechace el caso. Cada caso muestra a su vez detalles como su descripción, estado actual del caso, Tiempo que tiene en espera, Datos del médico que creó el caso y del Paciente que tiene asociado dicho caso, opiniones y archivos que contenga el mencionado caso.

### **Cliente**

En la interfaz “Casos Nuevos” se implementó una lista desplegable en JQueryMobile que nos permite ver un resumen del contenido y cuando la seleccionamos se expande mostrando todo el detalle a continuación. Esto permite incluir gran cantidad de ítems en forma de lista desplegable o acordeón en una misma pantalla ideal para dispositivos móviles sin necesidad de interactuar con diferentes pantallas para obtener la información.

### **Servidor**

La interfaz del usuario desplegada “Casos Nuevos” contiene las variables de sesión del Médico autenticado actual. La interfaz se comunica en Backend con el controlador “usuarioCasos.php” a quien le manda las variables de sesión de dicho médico, y a su vez se comunica por medio de la clase “class.php” con la base de datos para realizar la consulta por medio de la sentencia SQL y así encontrar todos los casos que tiene asociado dicho médico especialista.

Esta consulta retorna una serie de datos que son recorridos en el código “PHP” y son mostrados en la interfaz nuevamente devolviéndole así el control a la misma.

En la interfaz gráfica que muestra el detalle del caso se tiene la posibilidad de acceder a diferentes funcionalidades las cuales son:

- “Emitir Opinión”: el cual es un formulario con el botón “Enviar” que permite dar respuesta al caso que estamos viendo en ese momento.
- “Aceptar Caso” : el cuál es un botón que permite aceptar el caso y que quede asociado al Médico Especialista que está autenticado. (Esta funcionalidad será explicada a continuación)
- “Rechazar Caso”: el cuál es un botón que permite rechazar el caso y que sea devuelto a triaje para que sea reasignado. (Esta funcionalidad será explicada a continuación)
- Ver Opiniones
- Ver Archivos

La interfaz gráfica de esta funcionalidad la pueden encontrar en la figura 31:

Figura 31. Pantalla Casos Nuevos

- **Aceptar Casos**

Esta funcionalidad permite a un médico especialista aceptar el caso que está consultando para que quede asociado a su perfil y pueda ser respondido luego.

Para ello debe presionar el botón “Aceptar” disponible en el detalle del caso para enviar la solicitud al servidor.

Del lado del servidor al momento de aceptar un caso se cambia el estatus del mismo en la Base de datos de “Asignado” a estatus “En proceso 1er Nivel” y se actualizan tanto la tabla casos para asociarlo al médico especialista, así como la tabla historial para actualizar el historial de ejecución que nos permitirá llevar un control de todas las transacciones realizadas sobre ese caso en particular y sobre todo el sistema.

- **Rechazar Casos**

Esta funcionalidad permite a un médico especialista rechazar el caso que está consultando y que le habría sido asignado previamente por el médico de triaje.

Para ello debe presionar el botón “Rechazar” disponible en el detalle del caso para enviar la solicitud al servidor.

Del lado del servidor al momento de rechazar un caso se cambia el estatus del mismo en la Base de datos de “Asignado” a estatus “Rechazado 1er Nivel” y se actualizan tanto la tabla casos para modificar el estatus y enviarlo nuevamente a triaje, así como la tabla historial para actualizar el historial de ejecución que nos permitirá llevar un control de todas las transacciones realizadas sobre ese caso en particular y sobre todo el sistema.

- **Emitir Opinión**

Permite al usuario médico especialista emitir una opinión sobre un caso en particular.

## **Cliente**

Desde el punto de vista de la interfaz de usuario se diseñó la pantalla de emisión de una opinión que le permita así resolver el caso. Para ello tenemos los siguientes campos: “Título de Opinión”, “Cuerpo de la opinión”, “Adjuntar” y un botón “Enviar”.

En el campo de texto “Cuerpo de la Opinión” aprovechamos las bondades que nos brinda JQueryMobile que permite que ese campo sea tan extenso como se necesite redimensionándose así al tamaño de la pantalla haciendo más cómodo el ingreso de datos desde cualquier dispositivo móvil.

Para el campo adjuntar se implementó la posibilidad de cargar un archivo desde cualquier directorio del sistema y adjuntarlo para ser enviado vía formulario gracias al atributo `enctype="multipart/form-data"` que nos permite enviar archivos vía protocolo http específicamente por el método POST.

## **Servidor**

Del lado del servidor se toma la petición del usuario en el archivo “ctrlOpinion.php” quien es el controlador y va a obtener todos los datos de la petición.

En este controlador “ctrlOpinion.php” se obtiene el archivo enviado y se almacena en la variable `$_FILES` la cual es una Variables de Carga de Archivos HTTP que nos permitirá manipular dicho archivo y codificarlo en nuestro caso en un arreglo de Bytes para almacenarlo en la base de datos en un campo tipo BLOB.

Una vez hecho esto se almacenan los datos enviados y se devuelve el control a la interfaz.

- **Casos Por Resolver**

Permite consultar los casos que fueron aceptados por el médico especialista y que están a la espera de su respuesta.

Cada caso muestra a su vez detalles como su descripción, estado actual del caso, Tiempo que tiene en espera, Especialidad a la que fue asignado, Especialista que está atendiendo el caso, Nombre Medico Atención Primaria que creó el caso y datos del Paciente que tiene asociado dicho caso, opiniones y archivos que contenga el mencionado caso.

En la interfaz gráfica que nos muestra el detalle del caso se tiene la posibilidad de acceder a diferentes funcionalidades las cuales son:

- “Emitir Opinión” el cual es un formulario con el botón “Enviar” que permite dar respuesta al caso que se está consultando en ese momento.
- Ver Opiniones
- Ver Archivos

- **Archivo de Casos**

Permite consultar el histórico de los casos asociados al médico especialista autenticado que se encuentran en estatus “Resuelto 1er Nivel”, “Cerrados” o “Rechazados 1er Nivel” por el especialista.

Cada caso muestra a su vez detalles como su descripción, estado actual del caso, Tiempo que tiene en espera, Datos del Paciente que tiene asociado dicho caso, opiniones y archivos que contenga el mencionado caso.

En la interfaz gráfica que nos muestra el detalle del caso se tiene la posibilidad de acceder a diferentes funcionalidades las cuales son:

- Ver Opiniones
- Ver Archivos

### **Módulo Médico Rural**

Este módulo esta desarrollado para el usuario Médico Rural, mediante el cual el usuario puede ejecutar las diferentes funcionalidades dentro del sistema. Además de las funcionalidades generales que posee el usuario, también se encuentra:

- **Crear Caso**

Permite al usuario registrar un nuevo caso para que pueda ser tramitado y obtener la opinión de especialistas.

### **Cliente**

Desde el punto de vista de la interfaz de usuario se diseñó la pantalla de creación de casos que le permita al médico rural crear un nuevo caso. Para ello tenemos los siguientes campos: “Descripción”, “Centro Asistencial”, “Adjuntar” y un botón “Crear Caso”. Ver figura 32.

SOS Referencias Médicas  
Telemedicina para Venezuela Universidad Central de Venezuela. Facultad de Medicina

Vanessa Rodríguez  
[Cerrar Sesión](#)  
14-05-2014 19:04:22

### Médico Rural

Crear Casos

Descripción

Descripción / Motivo de la consulta

Centro Asistencial

Ambulatorio CIM UCV

Adjuntar: Seleccionar archivo Ningún archivo seleccionado

Crear Caso

Regresar

sosteledicina@ucv.ve

Figura 32. Nuevo Registro “Crear Caso”

En el campo de texto “Descripción” aprovechando las bondades que brinda JQueryMobile permite que ese campo sea tan extenso como se necesite redimensionándose así al tamaño de la pantalla haciendo más cómodo el ingreso de datos desde cualquier dispositivo móvil.

Para el campo “Centro Asistencial” se realizó una lista desplegable que contiene los centros ambulatorios disponibles para esa unidad de triaje.

Para el campo adjuntar se implementó la posibilidad de cargar un archivo desde cualquier directorio del sistema y adjuntarlo para ser enviado vía formulario gracias al atributo `enctype="multipart/form-data"` que nos permite enviar archivos vía protocolo http específicamente por el método POST.

## Servidor

Del lado del servidor se toma la petición del usuario en el archivo “ctrlOpinion.php” quien es el controlador y va a obtener todos los datos de la petición.

En este controlador “ctrlOpinion.php” se obtiene el archivo enviado y se almacena en la variable `$_FILES` la cual es una Variables de Carga de Archivos HTTP que nos permitirá manipular dicho archivo y codificarlo en nuestro caso en un arreglo de Bytes para almacenarlo en la base de datos en un campo tipo BLOB. Una vez hecho esto se almacenan los datos enviados y se devuelve el control a la interfaz.

- **Consultar Casos**

Permite consultar los casos que fueron creados por el Médico Rural autenticado en el sistema. Esta funcionalidad se divide en tres grandes rubros:

Casos Abiertos: todos los casos que fueron creados por el médico rural y que aún no han sido resueltos. Muestra los casos que tienen estatus “En espera”, “Asignado”, “En Proceso 1er nivel”, “En proceso 2do nivel”, “Rechazado 1er nivel” y “Rechazado 2do nivel”.

Casos SOS: todos los casos que fueron resueltos por Especialistas, es decir que se encuentran en estatus “Resuelto 1er Nivel” y Resuelto 2do Nivel”.

Casos Cerrados: todos los casos que fueron cerrados por el Médico Rural y por ello se encuentran en estatus “Cerrado”.

Cada caso muestra a su vez detalles como su descripción, estado actual del caso, Tiempo que tiene en espera, Especialidad a la que fue asignado, Especialista que está atendiendo el caso, Nombre Medico Atención Primaria que creó el caso y datos del Paciente que tiene asociado dicho caso, opiniones y archivos que contenga el mencionado caso.

En la interfaz gráfica que nos muestra el detalle del caso se tiene la posibilidad de acceder a diferentes funcionalidades las cuales son:

- Ver Opiniones
- Ver Archivos
- Cerrar Casos: Es un botón que permite cerrar el caso que se está consultando. Esta funcionalidad será explicada a continuación.

- **Cerrar Casos**

Esta funcionalidad permite a un médico rural cerrar el caso que está consultando.

Para ello debe presionar el botón “Cerrar” disponible en el detalle del caso para enviar la solicitud al servidor.

Del lado del servidor al momento de cerrar un caso se cambia el estatus del mismo en la Base de datos a estatus “Cerrado” y se actualizan tanto la tabla casos para modificar el estatus, así como

la tabla historial para actualizar el historial de ejecución que nos permitirá llevar un control de todas las transacciones realizadas sobre ese caso en particular y sobre todo el sistema.

La interfaz gráfica de Cerrar Casos la pueden encontrar en la figura 33.



Figura 33. Pantalla Cerrar Casos

#### 4.4.4. Etapa IV. Pruebas

A continuación se muestran el conjunto de pruebas funcionales de tipo caja cerrada que se realizaron. Se destaca que generalmente se recomienda que el equipo de desarrollo no sea el mismo que realice las pruebas sobre las funcionalidades del sistema, es por eso que en las actividades de pruebas se contó con la participación del equipo del Centro de Informática Médica.

A continuación se describen las pruebas realizadas en un total de 14 Tablas.

Tabla 6. Prueba de funcionamiento 1

|                                 |   | PRUEBAS        |          |
|---------------------------------|---|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Inicio de sesión</b>   | <b>Prueba:</b> | <b>1</b> |
| <b>Objetivo:</b>                | Constatar el inicio de sesión del usuario médico de triaje, médico especialista y médico rural.   |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>Se ingresa para cada uno de los roles del sistema, el nombre de usuario y contraseña.</li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria   |                |          |
| <b>Resultado Obtenido:</b>      | El usuario accede al sistema y puede realizar acciones según su perfil.   |                |          |
| <b>Observaciones:</b>           | Se constata que efectivamente cada uno de los tipos de usuarios ingresa al sistema de manera correcta.                                  |                |          |

Tabla 7. Prueba de funcionamiento 2

|                                 |  | PRUEBAS        |          |
|---------------------------------|--|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Creación de un nuevo caso</b>   | <b>Prueba:</b> | <b>2</b> |
| <b>Objetivo:</b>                | Registrar un nuevo caso en la base de datos del sistema.   |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>Se ingresan los datos para la creación del caso <ul style="list-style-type: none"> <li>Descripción del caso</li> <li>Centro Asistencial</li> <li>Archivo Adjunto (Opcional)</li> </ul> </li> <li>Se pasa el Id del médico y la fecha actual mediante una consulta interna.</li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria  |                |          |
| <b>Resultado Obtenido:</b>      | Visualización del Caso Creado en la base de datos del sistema con los datos que coinciden con los ingresados   |                |          |
| <b>Observaciones:</b>           |  |                |          |

**Tabla 8. Prueba de funcionamiento 3**

|                                 |  | <b>PRUEBAS</b> |          |
|---------------------------------|--|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Consulta de casos - Médico Rural</b>  | <b>Prueba:</b> | <b>3</b> |
| <b>Objetivo:</b>                | Ver el detalle de los casos registrados en el sistema según el estatus seleccionado desde el perfil Médico Rural.  |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona el tipo de caso a ser consultado                             <ul style="list-style-type: none"> <li>○ Casos Pendientes</li> <li>○ Casos Resueltos</li> <li>○ Casos Cerrados</li> <li>○ Se pasa el Id del médico y la fecha actual mediante una consulta interna.</li> </ul> </li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria  |                |          |
| <b>Resultado Obtenido:</b>      | Visualización del detalle del caso seleccionado según su estatus en sistema.   |                |          |
| <b>Observaciones:</b>           | Se presentaron problemas para asociar los casos creados por el médico rural a su perfil para consultar el detalle de los mismos. Este problema fue resuelto satisfactoriamente.  |                |          |

**Tabla 9. Prueba de funcionamiento 4**

|                                 |   | <b>PRUEBAS</b> |          |
|---------------------------------|---|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Consulta de casos - Médico Especialista</b>  | <b>Prueba:</b> | <b>4</b> |
| <b>Objetivo:</b>                | Ver el detalle de los casos registrados en el sistema según el estatus seleccionado desde el perfil Médico Especialista.  |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona el tipo de caso a ser consultado                             <ul style="list-style-type: none"> <li>○ Casos Nuevos</li> <li>○ Casos Por Resolver</li> <li>○ Archivo de Casos</li> </ul> </li> <li>• Se pasa el Id del médico y la fecha actual mediante una consulta interna.</li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria   |                |          |
| <b>Resultado Obtenido:</b>      | Visualización del detalle del caso seleccionado según su estatus en sistema.  |                |          |
| <b>Observaciones:</b>           |   |                |          |

**Tabla 10. Prueba de funcionamiento 5**

|                                 |   | PRUEBAS        |          |
|---------------------------------|---|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Consulta de casos - Médico Triage</b>  | <b>Prueba:</b> | <b>5</b> |
| <b>Objetivo:</b>                | Ver el detalle de los casos registrados en el sistema según el estatus seleccionado desde el perfil Médico Triage.  |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona el tipo de caso a ser consultado                             <ul style="list-style-type: none"> <li>○ Casos Nuevos</li> <li>○ Casos Asignados</li> <li>○ Casos Resueltos</li> <li>○ Casos Cerrados</li> </ul> </li> <li>• Se pasa el Id del médico y la fecha actual mediante una consulta interna.</li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria   |                |          |
| <b>Resultado Obtenido:</b>      | Visualización del detalle del caso seleccionado según su estatus en sistema.  |                |          |
| <b>Observaciones:</b>           |   |                |          |

**Tabla 11. Prueba de funcionamiento 6**

|                                 |  | PRUEBAS        |          |
|---------------------------------|--|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Aceptar un Caso</b>   | <b>Prueba:</b> | <b>6</b> |
| <b>Objetivo:</b>                | Aceptar un caso asignado a un determinado médico especialista.   |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona el caso a ser aceptado</li> <li>• Se presiona el botón “aceptar”.</li> <li>• Se pasa el Id del médico y la fecha actual mediante una consulta interna.</li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria  |                |          |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>• El caso queda en estatus “En proceso 1er Nivel” en la base de datos.</li> <li>• Visualización del caso aceptado asociado al médico especialista.</li> </ul>                     |                |          |
| <b>Observaciones:</b>           |  |                |          |

Tabla 12. Prueba de funcionamiento 7

|                                 |  | PRUEBAS        |          |
|---------------------------------|--|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Rechazo de un caso</b>  | <b>Prueba:</b> | <b>7</b> |
| <b>Objetivo:</b>                | Rechazar un caso asignado a un determinado médico especialista.  |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona el caso a ser aceptado</li> <li>• Se selecciona el botón “Rechazar”</li> <li>• Se pasa el Id del médico y la fecha actual mediante una consulta interna.</li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria.   |                |          |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>• El caso queda rechazado por el médico asociado y su estatus es “Rechazado 1er nivel”</li> <li>• Visualización del caso rechazado asociado al médico especialista.</li> </ul>      |                |          |
| <b>Observaciones:</b>           |  |                |          |

Tabla 13. Prueba de funcionamiento 8

|                                 |  | PRUEBAS        |          |
|---------------------------------|--|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Cierre de un caso</b>   | <b>Prueba:</b> | <b>8</b> |
| <b>Objetivo:</b>                | Cerrar un caso creado previamente por el médico rural.   |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona el caso a ser cerrado.</li> <li>• Se selecciona el botón “Cerrar”</li> <li>• Se pasa el Id del médico y la fecha actual mediante una consulta interna.</li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria.   |                |          |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>• El caso queda cerrado por el médico rural asociado y su estatus es “Cerrado”</li> <li>• Visualización del caso cerrado asociado al médico especialista.</li> </ul>              |                |          |
| <b>Observaciones:</b>           |  |                |          |

Tabla 14. Prueba de funcionamiento 9

|                                 |  | PRUEBAS        |          |
|---------------------------------|--|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Asignación de un caso</b>   | <b>Prueba:</b> | <b>9</b> |
| <b>Objetivo:</b>                | Asignar un caso que no ha sido resuelto, a un médico especialista.   |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona el caso a ser asignado.</li> <li>• Se selecciona el caso a asignar</li> <li>• Se selecciona la especialidad y médico asociado</li> <li>• Se presiona el botón “Asignar”.</li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria.   |                |          |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>• El caso queda cerrado por el médico rural asociado y su estatus es “Cerrado”</li> <li>• Visualización del caso cerrado asociado al médico especialista.</li> </ul>                                |                |          |
| <b>Observaciones:</b>           | Se presentaron problemas en la implementación de las listas seleccionables de Especialidades y Especialistas. Se resolvieron satisfactoriamente estos problemas implementando la funcionalidad utilizando Ajax.                            |                |          |

Tabla 15. Prueba de funcionamiento 10

|                                 |  | PRUEBAS        |           |
|---------------------------------|--|----------------|-----------|
| <b>Nombre de la prueba:</b>     | <b>Emitir Opinión</b>  | <b>Prueba:</b> | <b>10</b> |
| <b>Objetivo:</b>                | Emitir una opinión correspondiente al caso seleccionado.   |                |           |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona el caso al cuál se le emitirá una opinión.</li> <li>• Se ingresan los datos para emitir la opinión <ul style="list-style-type: none"> <li>○ Título de Opinión</li> <li>○ Cuerpo de la Opinión</li> <li>○ Archivo Adjunto (Opcional)</li> </ul> </li> <li>• Se pasa el Id del médico y la fecha actual mediante una consulta interna.</li> <li>• Se presiona el botón “Enviar”.</li> </ul> |                |           |
| <b>Evaluación de la prueba:</b> | Satisfactoria.   |                |           |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>• El caso queda en estatus “Resuelto 1er Nivel”.</li> <li>• Se genera la opinión en base de datos asociada al caso.</li> <li>• Visualización del caso y la opinión emitida con los datos que coinciden con los ingresados</li> </ul>  |                |           |
| <b>Observaciones:</b>           |  |                |           |

Tabla 16. Prueba de funcionamiento 11

|                                 |  | PRUEBAS        |           |
|---------------------------------|--|----------------|-----------|
| <b>Nombre de la prueba:</b>     | <b>Cargar Imagen</b>   | <b>Prueba:</b> | <b>11</b> |
| <b>Objetivo:</b>                | Cargar una imagen asociada a un caso y almacenarla en la base de datos del sistema.  |                |           |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona la imagen que se desea adjuntar.</li> <li>• El sistema valida el tamaño y el formato de la imagen.</li> <li>• Se presiona el botón “Adjuntar”.</li> </ul> |                |           |
| <b>Evaluación de la prueba:</b> | Satisfactoria.   |                |           |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>• Se almacena la imagen en la base de datos del sistema.</li> </ul>   |                |           |
| <b>Observaciones:</b>           | Se presentaron inconvenientes al momento de cargar las imágenes al sistema. Se resolvieron los problemas de forma satisfactoria.   |                |           |

Tabla 17. Prueba de funcionamiento 12

|                                 |  | PRUEBAS        |           |
|---------------------------------|--|----------------|-----------|
| <b>Nombre de la prueba:</b>     | <b>Ver Imagen adjunta</b>  | <b>Prueba:</b> | <b>12</b> |
| <b>Objetivo:</b>                | Visualizar todas las imágenes que tiene asociado un caso .   |                |           |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona el caso a consultar.</li> <li>• Se selecciona el nombre de la imagen que se desea consultar.</li> </ul> |                |           |
| <b>Evaluación de la prueba:</b> | Satisfactoria.   |                |           |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>• Se despliega en pantalla la imagen seleccionada asociada al caso</li> </ul>   |                |           |
| <b>Observaciones:</b>           |  |                |           |

Tabla 18. Prueba de funcionamiento 13

|                                 |   | PRUEBAS        |           |
|---------------------------------|---|----------------|-----------|
| <b>Nombre de la prueba:</b>     | <b>Ver Opiniones del Caso</b>   | <b>Prueba:</b> | <b>13</b> |
| <b>Objetivo:</b>                | Visualizar todas las opiniones asociadas a un caso.   |                |           |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>Se selecciona el caso a consultar.</li> <li>Se selecciona el nombre de la opinión que se desea consultar.</li> <li>Se seleccionan los botones que indican que existen más opiniones a mostrar</li> </ul> |                |           |
| <b>Evaluación de la prueba:</b> | Satisfactoria.  |                |           |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>Se despliega en pantalla la opinión seleccionada asociada al caso.</li> <li>Se despliegan unos botones que indican si existen más opiniones asociadas al caso</li> </ul>                                 |                |           |
| <b>Observaciones:</b>           | El cliente no estuvo de acuerdo con la interfaz diseñada. Se rediseñó la interfaz de muestra de las opiniones y se solventó satisfactoriamente.   |                |           |

Tabla 19. Prueba de funcionamiento 14

|                                 |   | PRUEBAS        |           |
|---------------------------------|---|----------------|-----------|
| <b>Nombre de la prueba:</b>     | <b>Salir del Sistema</b>  | <b>Prueba:</b> | <b>14</b> |
| <b>Objetivo:</b>                | Salir del sistema de forma controlada   |                |           |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>Se selecciona el botón “Cerrar Sesión”</li> </ul>  |                |           |
| <b>Evaluación de la prueba:</b> | Satisfactoria.  |                |           |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>Se cierra sesión del usuario borrando los datos de sesión almacenados.</li> <li>Se visualiza la pantalla de inicio indicando que se ha desconectado de forma exitosa.</li> </ul> |                |           |
| <b>Observaciones:</b>           |   |                |           |

## **4.5. Etapas de desarrollo aplicación nativa en Android**

En este apartado se describen el conjunto de procedimientos llevados a cabo durante cada etapa de desarrollo del software de la aplicación nativa desarrollada en el sistema operativo móvil Android.

### **4.5.1. Etapa I. Planificación**

Una vez realizados los diferentes estudios de factibilidad y las investigaciones acerca de las funcionalidades requeridas, además de la aplicación Web móvil, se planteó el desarrollo de una aplicación nativa implementada en el sistema operativo móvil Android, debido a los beneficios que ofrece, pudiendo así, implementar el sistema y abarcar mayores funciones como las que ofrece una aplicación nativa.

Entre las ventajas analizadas que representa desarrollar una aplicación nativa en este sistema operativo, se encuentra la principal, permite el uso de la aplicación bajo un ambiente «*stand alone*», en donde el usuario, sin restricción del uso de Internet, realiza las diferentes actividades propias de su perfil, además, se encuentra la funcionalidad de agregar contenido multimedia (imagen, audio, etc.), propia de las aplicaciones en estos dispositivos.

Para esta aplicación los requerimientos específicos se basan en un sólo usuario: médico rural, con acceso al sistema desde el lugar en donde se encuentre, y permitiendo que el mismo pueda hacer uso de la aplicación sin conexión a Internet, aunque los datos sólo se enviarán en el momento que el dispositivo tenga conexión y sincronice la comunicación.

### **4.5.2. Etapa II. Diseño**

Para el análisis de la aplicación se utilizó UML, que permitió visualizar, especificar, construir y documentar el sistema, además es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos.

Los modelos además, al ser una representación que incluye todos los detalles de los originales, permitieron probar más fácilmente el sistema que se modela y determinar los errores.

A través de éste lenguaje se desarrollaron los diagramas del sistema tal como lo son: diagramas de casos de uso, diagrama de secuencia y diagramas de clases, mediante los cuales se especifica el funcionamiento del sistema y la interacción de los diferentes usuarios con el mismo.

- **Diagrama de casos de uso**

Como se mencionó anteriormente, a través de la notación gráfica UML se representan los casos de uso. Para el desarrollo de esta aplicación el diagrama de casos de uso consta de los siguientes elementos:

- **Actores:** los diferentes actores que interactúan con la aplicación son: médico rural.
- **Casos de Uso:** las actividades que el usuario ejecuta en el sistema. (Ingresar al sistema, ver casos, crear caso, etc.). Se representan a través de diferentes niveles de detalle: Nivel 0 (figura 34), nivel 1 (figura 35).
- **Relaciones de Uso, Herencia y Comunicación:** se utilizan las tres relaciones principales entre los casos de uso son soportadas por el estándar UML: Inclusión (*include*), Extensión (*extend*) y generalización.



Figura 34. Diagrama de casos de uso nivel 0.

En la tabla 20, se puede apreciar los casos de uso que conforman el nivel 0 del diseño de la aplicación nativa Android, así como sus respectivas descripciones.

**Tabla 20. Casos de uso nivel 0. Android**

| Casos de Uso Nivel 0             |  |
|----------------------------------|--|
| <b>Pre-condición</b>             | Ninguna.   |
| <b>Nombre</b>                    | Ingresar al Sistema.   |
| <b>Actores</b>                   | Todos los usuarios (Médico Triage, Médico Especialista y Médico Rural)   |
| <b>Descripción</b>               | El usuario debe ingresar los datos de acceso al sistema, para poder ser validados y así realizar las diferentes funcionalidades según su perfil. |
| <b>Casos de uso relacionados</b> | Ninguno  |
| <b>Entrada</b>                   | Datos del usuario  |
| <b>Salida</b>                    | Ingreso correcto al sistema  |

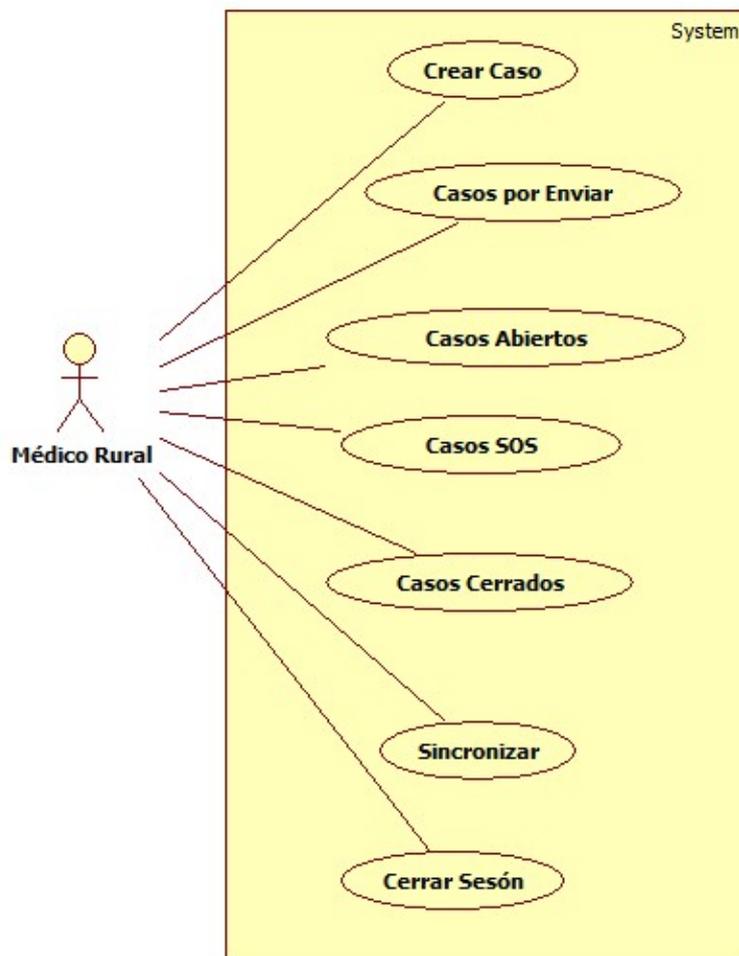


Figura 35. Diagrama de casos de uso nivel 1.

En la tabla 21, se detallan los casos de uso, sus descripciones correspondientes y casos asociados, que conforman el nivel 1 de detalle de la aplicación nativa en Android.

**Tabla 21. Casos de uso nivel 1. Android**

| <b>Casos de Uso Nivel 1</b>      |  |
|----------------------------------|--|
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Crear Caso.  |
| <b>Actores</b>                   | Médico Rural.  |
| <b>Descripción</b>               | El usuario puede crear un caso nuevo dentro del sistema.   |
| <b>Casos de uso relacionados</b> | Ninguno  |
| <b>Entrada</b>                   | Datos del nuevo caso a crear, descripción, archivos adjuntos y selección de centro asistencial.      |
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Casos de Uso Nivel 1</b>      |  |
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Casos por Enviar   |
| <b>Actores</b>                   | Médico Rural.  |
| <b>Descripción</b>               | El usuario puede visualizar todos los casos creados y que aun no han sido enviados.                  |
| <b>Casos de uso relacionados</b> | Ninguno  |
| <b>Entrada</b>                   | Ninguna  |
| <b>Salida</b>                    | Todos los casos pendientes por enviar que tiene asociado el usuario correspondiente.                 |
| <b>Casos de Uso Nivel 1</b>      |  |
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Casos Abiertos   |
| <b>Actores</b>                   | Médico Rural.  |
| <b>Descripción</b>               | El usuario puede visualizar todos los casos que aun se encuentran a la espera de una opinión médica. |
| <b>Casos de uso relacionados</b> | Ninguno  |
| <b>Entrada</b>                   | Ninguna  |
| <b>Salida</b>                    | Todos los casos que no tienen opinión médica emitida.  |

**Tabla 21.1. Casos de uso nivel 1. Android (Continuación)**

| <b>Casos de Uso Nivel 1</b>      |  |
|----------------------------------|--|
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Casos SOS  |
| <b>Actores</b>                   | Médico Rural.  |
| <b>Descripción</b>               | El usuario puede visualizar todos los casos que han sido creados y se encuentran enviados al servidor de SOS-TRIAJE. |
| <b>Casos de uso relacionados</b> | Ninguno  |
| <b>Entrada</b>                   | Ninguna  |
| <b>Salida</b>                    | Todos los casos que cargados en el servidor de SOS-TRIAJE.   |
| <b>Casos de Uso Nivel 1</b>      |  |
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Casos Cerrados   |
| <b>Actores</b>                   | Médico Rural.  |
| <b>Descripción</b>               | El usuario puede visualizar todos los casos que han sido cerrados.   |
| <b>Casos de uso relacionados</b> | Ninguno  |
| <b>Entrada</b>                   | Ninguna  |
| <b>Salida</b>                    | Todos los casos cerrados en el sistema.  |
| <b>Casos de Uso Nivel 1</b>      |  |
| <b>Pre-condición</b>             | Haber ingresado correctamente al sistema.  |
| <b>Nombre</b>                    | Salir del Sistema.   |
| <b>Actores</b>                   | Todos los usuarios (Médico Triage, Médico Especialista y Médico Rural)   |
| <b>Descripción</b>               | Permite al usuario salir del sistema.  |
| <b>Casos de uso relacionados</b> | Ninguno  |
| <b>Entrada</b>                   | Ninguna.   |
| <b>Salida</b>                    | Sesión de usuario cerrada.   |

- **Diagrama de Clases**

El diagrama de clases se puede apreciar en la figura 19, en la etapa de diseño de la aplicación web.

- **Diagrama de Actividad**

Este diagrama muestra el comportamiento desde una actividad hacia otra actividad dentro de la aplicación, es decir, muestra el flujo de control general. Ver apéndice A, desde Figura A.2 – Figura A.9.

- **Máquina de Estados**

La máquina de estado, como se mencionó anteriormente, representan un conjunto de estados que atraviesa un objeto durante su tiempo de vida dentro del sistema móvil. Ver figura 21.

- **Modelo Entidad – Relación**

Este diagrama permite a través del modelado de datos representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades. Se diseñó el siguiente modelo para crear a partir de él, una base de datos que contenga todas las tablas y atributos necesarios para almacenar los datos que proporciona el módulo de Referencia Médicas y lograr con ello el manejo de manera óptima de los mismos. Ver Apéndice A, Figura A.10.

### **4.5.3. Etapa III. Codificación**

En el proceso de codificación se implementó la arquitectura propia del sistema móvil nativo basado en el sistema operativo Android y se desarrollaron los módulos y funcionalidades de cada usuario, a través de diferentes herramientas de software.

#### **4.5.3.1. Arquitectura**

La arquitectura en base a la cual se diseñó la aplicación nativa en Android está compuesta por:

- Base de Datos Local (SQLite), la cual almacena localmente todas las operaciones realizadas por el usuario mientras se encuentre fuera del alcance de Internet.
- Dispositivos móviles, los diferentes dispositivos móviles que utilicen el sistema operativo móvil Android.
- Servicios Web, que permiten la comunicación entre la aplicación y la base de datos de Referencias Médicas.

- Base de Datos Referencias Médicas (MySQL), la cual almacena todos los registros creados y sus opiniones y archivos asociados.

En la figura 36 se describe la arquitectura mediante la cual está estructurada la aplicación nativa:

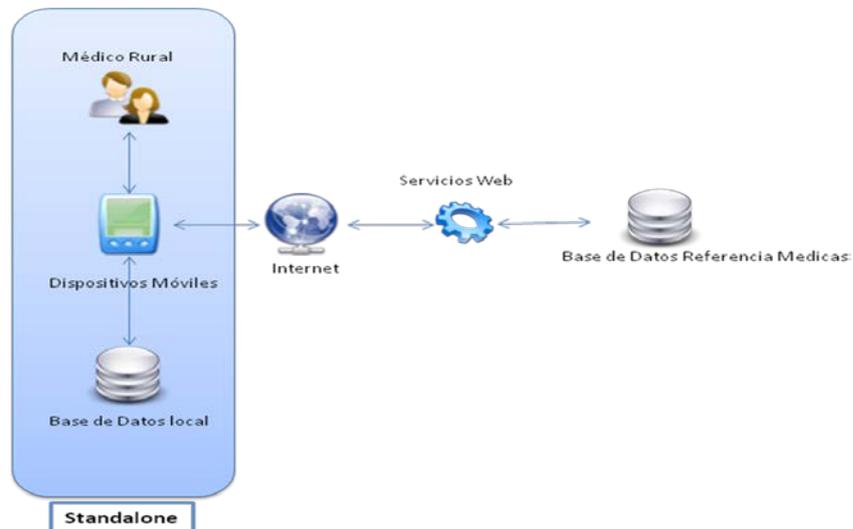


Figura 36. Arquitectura Sistema Móvil Nativo "Android"

#### 4.5.3.2. Sistema Operativo Android

Como se detallo en el Capítulo II, el sistema operativo móvil bajo el cual se desarrollo la aplicación nativa, es Android, un sistema operativo basado en el kernel de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes tabletas, inicialmente desarrollado por Android, Inc. Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android SDK (*Software Development Kit*), pero están disponibles otras herramientas de desarrollo, incluyendo un Kit de desarrollo nativo para aplicaciones o extensiones en C C++, Google App Inventor, un entorno visual para programadores novatos y varios marcos de aplicaciones basadas en la web multiteléfono. [26]

Para esta etapa del desarrollo de ésta aplicación en dicho sistema, se utilizaron las siguientes herramientas y lenguajes de programación:

### 4.5.3.3. Java

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, "*write once, run anywhere*"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de Web, con unos 10 millones de usuarios reportados. [27]

El lenguaje Java se creó con cinco objetivos principales:

- Debería usar el paradigma de la programación orientada a objetos.
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Debería incluir por defecto soporte para trabajo en red.
- Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
- Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

El sistema operativo móvil Android sólo utiliza la sintaxis y la semántica de Java, pero no incorpora en su totalidad las bibliotecas de clases de Java y APIs (*Application Programming Interface*) que acompañan a Java SE o ME. Sin embargo, hay diversas herramientas en el mercado de empresas como Myriad Group y UpOnTek que dan un servicio de conversión entre J2ME y Android.

### 4.5.3.4. Entorno de desarrollo Android SDK (Software Development Kit)

El SDK de Android representa un conjunto de herramientas de desarrollo utilizadas para crear aplicaciones para la plataforma Android. El SDK de Android incluye lo siguiente [28]:

- Bibliotecas necesarias
- Depurador
- Un emulador
- La documentación pertinente para las interfaces de programación de aplicaciones de Android (API)
- Código fuente de ejemplo

- Tutoriales para el sistema operativo Android

Las plataformas de desarrollo que compatibles con SDK incluyen sistemas operativos como Windows (XP o posterior), Linux (cualquier distribución Linux reciente) y Mac OS X (10.4.9 o posterior). Los componentes del SDK de Android se pueden descargar por separado. [28]

Aunque el SDK se puede utilizar para escribir programas basados en Android, el método más común es el uso de un entorno de desarrollo integrado (IDE). Dado que las aplicaciones de Android están escritas en Java, se debe tener el JDK (*Java Development Kit*) instalado.

#### **4.5.3.5. XML (eXtensible Markup Language)**

Es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos, para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información. [29]

XML ofrece una representación estructural de los datos que se puede implementar ampliamente y es fácil de distribuir. XML es un subconjunto de SGML optimizado para el Web, además garantiza que los datos estructurados sean uniformes e independientes de aplicaciones o fabricantes. La interoperabilidad resultante está creando rápidamente una nueva generación de aplicaciones de comercio electrónico en la Web. [29]

Un documento XML tiene dos estructuras, una lógica y otra física. Físicamente, el documento está compuesto por unidades llamadas entidades. Una entidad puede hacer referencia a otra entidad, causando que esta se incluya en el documento. Cada documento comienza con una entidad documento, también llamada raíz. Lógicamente, el documento está compuesto de declaraciones, elementos, comentarios, referencias a caracteres e instrucciones de procesamiento, todos los cuales están indicados por una marca explícita. Las estructuras lógica y física deben encajar de manera adecuada.

#### **4.5.3.6. JSON (Java Script Object Notation)**

Es un formato de intercambio de datos ligero. Se basa en un subconjunto del lenguaje de programación JavaScript y es completamente independiente del lenguaje pero utiliza convenciones que son familiares para los programadores de los lenguajes, C , C + +, C #, Java,

JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen de JSON un lenguaje ideal para el intercambio de datos. [30]

JSON se basa en dos estructuras [30]:

- Una colección de pares nombre / valor. En varios idiomas, esto se realiza como un objeto, registro, estructura, diccionario, tabla hash, lista con clave, o una matriz asociativa.
- Una lista ordenada de valores. En la mayoría de los idiomas, esto se realiza en forma de matriz, vector, lista o secuencia.

Se trata de estructuras de datos universales. Prácticamente todos los lenguajes de programación modernos los soportan de una forma u otra. Tiene sentido que un formato de datos que es intercambiable con los lenguajes de programación también se base en estas estructuras. [30]

Para escribir JSON debemos tener en cuenta lo siguiente:

La creación de los objetos JSON implica escribir datos, para ello:

- Los datos están separados por comas.
- los datos se escriban en pares, siendo primero el nombre o atributo del mismo y luego el valor del dato.
- Los objetos JSON están rodeados por llaves “{ }”
- Llaves cuadradas “[ ]” guardan arreglos, incluyendo otros objetos JSON

En la figura 37 se puede apreciar en código de creación de un objeto Json.

```
//Ejemplo de JSON
{
  "nombre":"Vanessa Rodriguez",
  "edad":25,
  "especialidad":["Traumatología","Fisiatría"]
}
```

Figura 37. Código de creación de objeto JSON

#### **4.5.3.7. SQLite**

Es un sistema de gestión de bases de datos relacional contenida en una relativamente pequeña biblioteca escrita en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp. [31]

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

Debido a su pequeño tamaño, SQLite es muy adecuado para los sistemas integrados, y también está incluido en [31]:

- Android
- BlackBerry
- Windows Phone 8
- Google Chrome
- iOS
- Maemo
- MeeGo
- Symbian OS
- webOS

#### **4.5.3.15. Servicios Web**

Un servicio web (*Web Service o Web services*) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes

de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. [32]

La principal razón para usar servicios Web es que se pueden utilizar con HTTP sobre TCP (*Transmission Control Protocol*) en el puerto 80. Dado que las organizaciones protegen sus redes mediante cortafuegos, *firewall*, que filtran y bloquean gran parte del tráfico de Internet, cierran casi todos los puertos TCP salvo el 80, que es, precisamente, el que usan los navegadores. Los servicios Web utilizan este puerto, por la simple razón de que no resultan bloqueados. Es importante señalar que los servicios web se pueden utilizar sobre cualquier protocolo, sin embargo, TCP es el más común.

Una tercera razón por la que los servicios web son muy prácticos es que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más utilizada.

A continuación se listan los tres estilos de usos más comunes [32]:

- **Llamadas a Procedimientos Remotos (*Remote Procedure Calls, RPC*):** Los Servicios Web basados en RPC presentan una interfaz de llamada a procedimientos y funciones distribuidas, lo cual es familiar a muchos desarrolladores. Típicamente, la unidad básica de este tipo de servicios es la operación WSDL (WSDL es un descriptor del Servicio Web, es decir, el homólogo del IDL para COM). Las primeras herramientas para Servicios Web estaban centradas en esta visión.
- **Arquitectura Orientada a Servicios (*Service-oriented Architecture, SOA*).** Los Servicios Web pueden también ser implementados siguiendo los conceptos de la arquitectura SOA, donde la unidad básica de comunicación es el mensaje, más que la operación. Esto es típicamente referenciado como servicios orientados a mensajes.

Los Servicios Web basados en SOA son soportados por la mayor parte de desarrolladores de software y analistas. Al contrario que los Servicios Web basados en RPC, este estilo es débilmente acoplado, lo cual es preferible ya que se centra en el “contrato” proporcionado por el documento WSDL, más que en los detalles de implementación subyacentes.

- **Representación de Estado de Transferencia (*Representation State Transfer, REST*).**

Los Servicios Web basados en REST intentan emular al protocolo HTTP o protocolos similares mediante la restricción de establecer la interfaz a un conjunto conocido de operaciones estándar (por ejemplo GET, PUT,...). Por tanto, este estilo se centra más en interactuar con recursos con estado, que con mensajes y operaciones.

En este trabajo se utilizaron servicios web basados en **REST** mejor conocidos como Servicios **RESTfull**.

**REST** es un estilo de arquitectura de software para sistemas distribuidos tales como la web, a diferencia de SOAP, se centra en el uso de los estándares HTTP y XML para la transmisión de datos sin la necesidad de contar con una capa adicional. Las operaciones (ó funciones) se solicitarán mediante GET, POST, PUT y DELETE, por lo que no requiere de implementaciones especiales para consumir estos servicios. Además se podrá utilizar JSON en vez de XML como contenedor de la información, por lo que será aconsejable utilizar este protocolo cuando busquemos mejorar el rendimiento, o cuando disponemos de escasos recursos, como sería el caso de los dispositivos móviles. [32]

Una implementación concreta de un servicio web REST sigue cuatro principios de diseño fundamentales:

- Utiliza los métodos HTTP de manera explícita
- No mantiene estado
- Expone URLs con forma de directorios
- Transfiere XML, JavaScript Object Notation (JSON), o ambos

#### **4.5.3.16. Implementación de Servicio Web REST desde Android**

A continuación se detalla la implementación del servicio web REST.

##### **Cliente**

Para lograr la comunicación con el servicio web desde la aplicación en Android, se implementó el API HttpClient que provee Apache. La principal función de ésta biblioteca es ejecutar métodos HTTP (GET, POST, PUT, HEAD,...)

La ejecución de un método HTTP implica el intercambio entre la aplicación y el servidor, de peticiones HTTP y respuestas HTTP, que normalmente son realizadas internamente por el HttpClient.

Para realizar una petición POST al servidor, se configura el cliente en el dispositivo Android de la siguiente manera:

- a. Se crea la función `httpPostRequest()`, ver figura 38, a la que se asigna como primer parámetro la URL a la que se le hará la petición (donde se encuentra alojado el servicio web) y como segundo parámetro se le pasan los valores que serán enviados al servidor.

```
public static String httpPostRequest(String url,ArrayList<NameValuePair> nameValuePairs) {
```

Figura 38. Función `httpPostRequest()`

- b. Dentro de la función, se crea el cliente `HttpPost` asignándole la URL en donde se hará la petición, luego se configuran los parámetros de la conexión (como el tiempo de *timeout* en caso de no haber respuesta en la conexión al servidor), se le asignan los parámetros a la petición y se ejecuta la consulta, como se puede apreciar en la figura 39. Esta consulta puede arrojar una excepción, por eso se maneja en el bloque *try – catch*.
- c. La función retorna un objeto `Json`, en donde se obtendrá la respuesta del servidor a nuestra petición.

```

// Crear una nueva petición http con Cabecera POST
HttpPost httppost = new HttpPost(url);

//Configuración de los parametros de la conexión
HttpParams params = new BasicHttpParams();
HttpConnectionParams.setConnectionTimeout(params, 15000);
HttpConnectionParams.setSoTimeout(params, 15000);
HttpClient httpclient = new DefaultHttpClient(params);

try {
//Se envia la consulta al servidor
httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));

// Execute HTTP Post Request
Log.i("url", url);
HttpResponse response = httpclient.execute(httppost);
if(response.getStatusLine().getStatusCode() == 200){
return inputStreamToString(response.getEntity().getContent());
}else{
return null;
}
} catch (ClientProtocolException e) {
// TODO Auto-generated catch block
e.printStackTrace();
return null;
} catch (Exception e) {
// TODO Auto-generated catch block
e.printStackTrace();
return null;
}
}

```

Figura 39. Objeto JSON

Para realizar una petición **GET** al servidor, se configura el cliente en el dispositivo Android de la siguiente manera:

- a. Se crea la función `httpGetRequest ()`, ver figura 40, a la que se le asigna como parámetros la URL a la que se le realiza la petición (donde se encuentra alojado el servicio web).

```
public static String httpGetRequest(String url) {
```

Figura 40. Función `httpGetRequest ()`

- b. Dentro de la función se crea el cliente `HttpGet` asignándole la URL en donde se hará la petición y luego se manda a ejecutar la consulta como se puede apreciar en la figura 41. Esta consulta puede arrojar una excepción, por eso es manejada en el bloque `try – catch`.

- c. La respuesta de esta consulta se almacena en un Arreglo de tipo Json para poder manipular los datos obtenidos desde el servicio web, parseando estos datos, como se explica en el apartado “Parseo de Datos con JSON”.

Ver detalle en el Apéndice B – Documentación Técnica.

```
try {  
    // Crear una nueva petición http con Cabecera GET  
    HttpClient httpClient = new DefaultHttpClient();  
    HttpGet httpGet = new HttpGet(url);  
    //Se envia la consulta al servidor  
    HttpResponse response = (HttpResponse)httpClient.execute(httpGet);  
    HttpEntity httpEntity = response.getEntity();  
    is = httpEntity.getContent();  
} catch (UnsupportedEncodingException e) {  
    e.printStackTrace();  
}
```

Figura 41. Función HttpGet()

## Servidor

En el servidor, se crean los APIs para el servicio web en lenguaje .php quienes serán los encargados de recibir y manejar todas las peticiones hechas por el cliente Android.

El servicio web con extensión .php alojado en el servidor, se obtienen los datos de la petición bien sea por la variable \$\_POST[' '] si la petición fue vía POST, o mediante la variable \$\_GET[`] si la petición fue mediante GET. También se puede utilizar la variable \$\_REQUEST[`], ver figura 42.

```
$centro_id = $_POST['centro_id'];  
$descripcion = $_POST['descripcion'];  
$fecha_inicio = $_POST['fecha_inicio'];
```

Figura 42. Datos de Petición al servidor

Con los datos obtenidos, se realizan las funciones requeridas, bien sea conectarse a la base de datos local para hacer una consulta, para realizar una inserción de una imagen, o cualquier otra funcionalidad definida previamente.

Una vez que se procesa la solicitud y se ejecuta el servicio web, se retorna un objeto de tipo JSON para que sea devuelto y pueda ser manejado según su respuesta, tal como muestra la figura 43.

```
$user = $db->casosUp($id_caso, $version, $centro_id, $descripcion,
                  $fecha_inicio, $fecha_solucion);
if ($user) {
    //Respuesta: True si la solicitud se proceso exitosamente
    $response["exito"] = "true";
    $response["error_msg"] = "";
    echo json_encode($response);
} else {
    //Respuesta: False si la solicitud no se proceso exitosamente
    $response["exito"] = "false";
    $response["error_msg"] = "Error insertando casos";
    echo json_encode($response);
}
```

Figura 43. Objeto JSON de retorno

#### 4.5.3.17. Manejo de Base de datos SQLite

Para que la aplicación pueda funcionar correctamente de manera *stand alone*, se contó con una base de datos local en la aplicación Android para manejar la persistencia de los datos sin necesidad de tener una conexión a Internet o hacia algún servidor remoto.

Esta base de datos local en la aplicación móvil se implementó mediante una base de datos SQLite.

La creación, actualización y conexión de la base de datos se realiza desde la aplicación a través de una clase que hereda de `SqliteOpenHelper`. Esta clase gestiona la creación de la base de datos la primera vez que se accede a la misma con el método `onCreate` y la actualización de la base de datos cuando se detecte un cambio de versión de la misma con el método `onUpdate`. Esta clase se llamó `MySqliteOpenHelper`.

En la clase `MySqliteOpenHelper`, ver figura 44, se crearon las definiciones de todas las tablas de nuestra base de datos local llamada `Rural.sql`

```

public class MySQLiteOpenHelper extends SQLiteOpenHelper {

    private static final String DATABASE_NAME = "Rural";
    private static final int DATABASE_VERSION = 1;

    //Definición de tablas
    public static class TablaCaso{
        public static String TABLA_CASO = "caso";
        public static String COLUMNA_ID = "id";
        public static String COLUMNA_VERSION = "version";
        public static String COLUMNA_centro_id = "centro_id";
        public static String COLUMNA_descripcion = "descripcion";
        public static String COLUMNA_fecha_inicio = "fecha_inicio";
        public static String COLUMNA_fecha_solucion = "fecha_solucion";
        public static String COLUMNA_id_casosos = "id_casosos";
        public static String COLUMNA_paciente_id = "paciente_id";
        public static String COLUMNA_status_id = "status_id";
    }
}

```

Figura 44. Clase MySQLiteOpenHelper. SQLite

Adicionalmente se implementaron las siguientes funciones:

- onCreate() para crear la base de datos con sus respectivas tablas. Ver figura 45.
- onUpgrade() para actualizar la base de datos cuando se detecte un cambio de versión de la misma. Ver figura 45.
- deleteAllData () para borrar toda la data de la base de datos
- deleteAutoincrement () que permite resetear el índice de los campos autoincrement de algunas tablas de la base de datos.

```

@Override
public void onCreate(SQLiteDatabase db) {
    // TODO Auto-generated method stub
    db.execSQL(TABLE_CASO);
    db.execSQL(TABLE_HISTORIAL);
    db.execSQL(TABLE_ARCHI_UP);
    db.execSQL(TABLE_MISCASOS);
    db.execSQL(TABLE_RESPUESTA);
    db.execSQL(TABLE_ARCHIVO);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // TODO Auto-generated method stub
    db.execSQL("DROP TABLE IF EXISTS "+TABLE_CASO);
    db.execSQL("DROP TABLE IF EXISTS "+TABLE_HISTORIAL);
    db.execSQL("DROP TABLE IF EXISTS "+TABLE_ARCHI_UP);
    db.execSQL("DROP TABLE IF EXISTS "+TABLE_MISCASOS);
    db.execSQL("DROP TABLE IF EXISTS "+TABLE_RESPUESTA);
    db.execSQL("DROP TABLE IF EXISTS "+TABLE_ARCHIVO);
}

```

Figura 45. Funciones SQLite

Para ingresar datos en la base de datos Rural.sql y para obtener datos de la misma, se creó la clase DataSource() en donde se implementó la clase SQLiteDatabase que permite tener acceso a la base de datos local creada y aplicar las funciones CRUD (Create, Read, Update y Delete) sobre la misma.

#### 4.5.3.18. Sincronización de la aplicación con el servidor

Para implementar la sincronización de la base de datos local en la aplicación Android con la base de datos y servicios web en el servidor, se utilizó la clase AsyncTask la cual permite ejecutar operaciones en segundo plano sin bloquear la interfaz gráfica del usuario.

La clase AsyncTask tiene cuatro etapas que nos facilitan su implementación las cuales son [33]:

- **onPreExecute():** se invoca en el hilo principal, es decir en la interfaz gráfica del usuario. Este paso se utiliza normalmente para configurar la tarea asíncrona que se está llamando y para mostrar una barra de progreso en la interfaz de usuario mientras se configura el subproceso llamado en segundo plano, tal como se aprecia en la figura 46.

```
@Override
protected void onPreExecute() {
    // TODO Auto-generated method stub
    super.onPreExecute();
    dialog.setTitle("Autenticación de Usuario");
    dialog.setMessage("Validando Datos...");
    dialog.setIndeterminate(false);
    dialog.setCancelable(true);
    dialog.show();
}
```

Figura 46. Función onPreExecute().

- **doInBackground(Params...):** se invoca en el subproceso en segundo plano inmediatamente después de que onPreExecute() termina de ejecutarse. En esta etapa se ejecutan todas las tareas del subproceso necesarias y debe retornar el objeto que se especificó en la definición de la función. En esta etapa se calcula el tiempo estimado que tomará la realización de la tarea, que deberá pasarse a la última etapa de todo el proceso llamada onPostExecute() junto con el objeto que retorna esta etapa. Ver figura 47.

```

@Override
protected JSONObject doInBackground(String... args) {
    UserFunctions userFunction = new UserFunctions();
    JSONObject json = userFunction.loginUser(args[0], args[1]);
    return json;
}

```

Figura 47. Función doInBackground ().

- **onProgressUpdate(Progress...):** éste método es utilizado para mostrar cualquier tipo de progreso en la interfaz de usuario mientras la tarea en segundo plano sigue ejecutándose.
- **onPostExecute(Result...):** se invoca después de que el proceso en segundo plano ha sido terminado, es decir, luego de que el método doInBackground() se ejecuta, el resultado devuelto se pasa a este método como parámetro. Aquí se implementan todas aquellas acciones que se deben ejecutar luego de que el subprocesso culminó como por ejemplo actualizar la interfaz de usuario que llamó al subprocesso. Ver figura 48.

```

protected void onPostExecute(JSONObject json_user) {
    try {
        if (json_user.getString("exito") != null) {

            dialog.setMessage("Cargando los Datos del Usuario");
            dialog.setTitle("Obteniendo Datos..");
            dialog.dismiss();
            intent = new Intent(context, MainSegundaOpinion.class);
            context.startActivity(intent);
            context.finish();
        }else{
            Toast.makeText(context, "Error de Autenticación. Intente nuevamente",
                Toast.LENGTH_SHORT).show();
            dialog.setTitle("Error de Autencación");
            dialog.setMessage("Usuario / Password Incorrectos");
            dialog.dismiss();
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

```

Figura 48. Función onPostExecute().

#### 4.5.3.19. Parseo de Datos con JSON

El parseo es un proceso de análisis y posterior transformación de un bloque en un lenguaje/estructura determinada en otro bloque de lenguaje/estructura diferente. [30]

Para realizar la comunicación entre la aplicación en Android y la base de datos remota (mediante servicios web) se utilizó JSON ya que permite codificar objetos, arrays o cualquier otra serie de datos en un string y posteriormente poder decodificarlo (parsearlos) para poder utilizarlos. JSON es más ligero y fácil de parsear que XML y en el desarrollo de la aplicación para dispositivos móviles, la eficiencia en el uso de los recursos es fundamental para el correcto desempeño de nuestra aplicación en términos de rendimiento, velocidad de intercambio de los datos, ahorro de la batería entre otros. [30]

JSON es especialmente útil para enviar información entre cliente y servidor de una forma muy sencilla, puesto que cada componente decodifica la información según le convenga, indistintamente del resto del sistema e independiente al lenguaje en que estén implementados tanto el cliente como el servidor (En nuestro caso Java-Android para el cliente y PHP para nuestro servicio web). [30]

Para lograr la correcta comunicación entre el cliente y el servidor de la aplicación se realizó del lado del cliente la clase `JSONParser.java` la cual implementa las clases `JSONArray` y `JSONObject`.

Cuando se lee un objeto `Json` desde el servidor, el mismo se almacena en un `JSONObject` en caso de que los datos transferidos vengan en un Arreglo unidimensional (Ver figura 49) o se almacena en un `JSONArray` que no es más que un array de `JSONObjects` para el caso en que los datos del servidor vengan como un arreglo multidimensional. Una vez obtenido el `JSONArray`, recorreremos uno a uno cada elemento `JSONObject` pudiendo acceder a sus atributos y procesándolos como se puede apreciar en la figura 50.

```
JSONObject jsonObj = new JSONObject(json);
```

Figura 49. Objeto Json

```

// Parseamos la respuesta obtenida del servidor a un objeto JSON
JSONObject jsonObject = new JSONObject(sb.toString());
JSONArray archivos = jsonObject.getJSONArray("archivos");

/ Recorremos el array con los elementos
for(int i = 0; i < archivos.length(); i++) {
    JSONObject archivo = archivos.getJSONObject(i);

    // Creamos el objeto Archivo
    Archivo ar = new Archivo();
    ar.setId(archivo.getInt("id"));
    ar.setVersion(archivo.getInt("version"));
    ar.setJsonIMG(archivo.getString("adjunto"));
    ar.setCasoId(archivo.getInt("caso_id"));
    ar.setDescripcion(archivo.getString("descripcion"));
    ar.setNombre(archivo.getString("nombre"));

    // Almacenamos el objeto en el array que hemos creado anteriormente
    listaArchivo.add(ar);
}

```

Figura 50. JSONArray ()

#### 4.5.3.20. Funcionalidades Aplicación Android

En este apartado se describen las funcionalidades por las que está compuesta la aplicación Android.

- **Pantalla de Bienvenida (*Splash Screen*)**

Esta interfaz es generalmente una imagen que aparece en la aplicación apenas se inicia y que estará desplegada mientras se cargan todos los datos y se configura la aplicación en el dispositivo. Sirve para indicarle al usuario que se está iniciando la aplicación y que debe esperar unos instantes mientras este proceso es ejecutado.

La pantalla de bienvenida se desarrolló utilizando la clase SplashScreenActivity.java, la cual permite demorar o retardar el inicio de la aplicación utilizando la clase TimerTask y mientras se cargan los datos mostrar el logo de SOS TELEMEDICINA en el fondo de la pantalla como se puede apreciar en la figura 51.



Figura 51. Pantalla de Bienvenida (Splash Screen)

Como se mencionó anteriormente en el Capítulo II, en el archivo de configuración de la aplicación llamado `AndroidManifest.xml`, se debe configurar cuál es la actividad o *activity* inicial de la aplicación. Esta debe ser declarada de tipo LAUNCHER tal como se puede apreciar en la figura 52.

```
<activity
    android:name="com.teg.segundaopinion.SplashScreenActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity
```

Figura 52. Actividad creada en el archivo `AndoridManifest.xml`

- **Ingresar al Sistema**

Permite al usuario que se está autenticando acceder al sistema, según su perfil. El desarrollo de ésta funcionalidad es el siguiente:

## Cliente

Desde el punto de vista de la interfaz de usuario se diseñó la interfaz de ingreso al sistema de forma tal que sea usable e intuitiva para el usuario final, implementada en ella dos campos: “Usuario” y “Contraseña” además de un botón “Ingresar”. Ver figura 53.

Se realizaron validaciones en la vista específicamente en la clase Login.java, implementando la función validaDatos() la cual permite validar que los campos “Usuario” y “Contraseña” no estén vacíos, retornando un mensaje de error invitando al usuario a completar todos los campos.



The image shows a mobile application interface for 'Referencias Médicas'. At the top, there is a status bar with signal strength, 3G, and a battery icon, and a time of 1:56. Below the status bar is a header with the text 'Referencias Médicas' and a logo for 'SOS Telemedicina para Venezuela'. The main content area contains two input fields: 'Usuario' and 'Contraseña'. Below the 'Contraseña' field is a button labeled 'Ingresar'. At the bottom of the form area, there is a dark grey button with the text 'Debe rellenar todos los campos'. The footer of the application is a yellow bar with the email address 'sostelemedicina@ucv.ve'.

Figura 53. Interfaz de Ingreso al Sistema

Cabe destacar que todas estas validaciones se realizan en la vista (cliente) lo que permite que la petición que se realizará hacia el servidor ya venga verificada y validada y así establecer parámetros de seguridad para evitar posibles ataques como inyección de código SQL en formularios por ejemplo.

Una vez que se validan los datos del lado del usuario, se realiza la petición vía protocolo HTTP hacia el servidor, presionando el botón “Ingresar”. Esta petición se envía hacia el servidor invocando a un servicio web, el cual será el encargado de manejar dicha solicitud.

## Servidor

Para invocar al servidor, desde el lado del cliente específicamente desde la clase Login.java se invoca a la clase TareaAsincrona.java (que hereda de la clase AsyncTask) específicamente al método TareaAsincrona.execute() con el cual se crea un hilo de ejecución en Background para lograr la comunicación con el servidor y validar los datos del usuario para poder autenticarse.

Esta clase TareaAsincrona.java invoca en el método doInBackground a la clase UserFunction.java específicamente al método userFunction.loginUser(args[0], args[1]); al cual se le pasarán los datos introducidos por el usuario en la interfaz (usuario y contraseña) para ser enviados al servidor.

Una vez dentro de la función loginUser, se crea un arreglo del tipo NameValuePair el cual es un arreglo de pares clave-valor que se le agregan los datos introducidos por el usuario, así como también la URL en donde está alojado el servicio web en el servidor y una etiqueta adicional llamada tag que será utilizada para identificar el tipo de consulta en el servicio web como se puede apreciar en la figura 54.

```
public JSONObject loginUser(String email, String password){
    // Building Parameters
    ArrayList<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();
    nameValuePairs.add(new BasicNameValuePair("tag", login_tag));
    nameValuePairs.add(new BasicNameValuePair("usuario", email));
    nameValuePairs.add(new BasicNameValuePair("password", password));
    JSONObject json = jsonParser.getJSONFromUrl(LoginURL, nameValuePairs);
    return json;
}
```

Figura 54. Objeto Json. Validación de usuario

Esta solicitud se envía al servidor y debe retornar un objeto del tipo JSON el cual debemos decodificar para verificar la respuesta.

El servicio web invocado se encuentra en el servidor en el directorio “Android\index.php”. Este servicio toma los campos enviados por el usuario por medio de las variables \$\_POST que es la manera como fueron enviados los datos en la petición del usuario y obtiene la información que deberá ser validada.

Con el valor del usuario y contraseña enviados, se invoca a la función getUserByEmailAndPassword (usuario, contraseña) la cual se comunica con la base de datos en el servidor y valida la coincidencia de los mismos con los almacenados en la base de datos del

Sistema SOS-TRIAJE y así retorna los datos del usuario autenticado en caso de ser efectiva la consulta, o nos retornará falso en caso de no ser efectiva.

Esta respuesta se codificará en un objeto `json_encode()` que retornará el objeto JSON al cliente desde donde se realizó la petición, en nuestro caso a la clase `TareaAsincrona.java`. Ver figura 55.

```

if ($tag == 'login') {
    // Request type is check Login
    $email = $_POST['usuario'];
    $password = $_POST['password'];

    $user = $db->getUserByEmailAndPassword($email, $password);
    if ($user != false) {
        $response["exito"] = "true";
        $response["id"] = $user["id"];
        $response["nombre"] = $user["login"];
        $response["correo"] = $user["password"];
        $response["rol"] = $user["rol"];
        echo json_encode($response);
    } else {
        $response["exito"] = "false";
        $response["id"] = "";
        $response["nombre"] = "";
        $response["correo"] = "";
        echo json_encode($response);
    }
}

```

Figura 55. Validación de Usuario

Una vez en la clase `TareaAsincrona.java`, se maneja la respuesta en formato JSON, (parseando la respuesta). La clase `JSONParser.java` transforma al tipo de dato `String` y hace que sean entendibles por la aplicación en Android.

Si la autenticación fue exitosa se toman los datos del usuario que retornó el servicio web y se almacenan en una variable del tipo `SharedPreferences`.

`SharedPreferences`, ver figura 56, nos permite almacenar datos o preferencias para ser compartidos entre otros métodos o actividades dentro de nuestra aplicación Android y que no se borran al cerrar la aplicación, es decir tienen persistencia.

Apoyados en esta se crea algo similar a una sesión de usuario almacenando el ID del médico, nombre y apellido, correo, una bandera que nos indica si está autenticado llamada "isLogin" y por último el rol de este médico.

```
SharedPreferences.Editor editor = preferences.edit();
// guardar clave valor del usuario
if((exito.equals("true"))&&(rol.equals("Rural"))){
    editor.putString("id", id);
    editor.putString("nombre", nombre);
    editor.putString("correo", correo);
    editor.putBoolean("isLogin", true);
    dialog.dismiss();
    intent = new Intent(context, MainSegundaOpinion.class);
    context.startActivity(intent);
    // Se quita el login del stack de activitys
    context.finish();
}
else{
    editor.putBoolean("isLogin", false);
    Toast.makeText(context, "Error de Autenticación. Intente nuevamente", Toast.LENGTH_SHORT).show();
    dialog.setTitle("Error de Autencación");
    dialog.setMessage("Usuario / Password Incorrectos");
    dialog.dismiss();
}
}

editor.commit();
```

Figura 56. Variable SahredPreferences

Una vez almacenados los datos del usuario, se retorna el control a la interfaz de usuario como se puede apreciar en la figura 57.

Si el usuario no se autenticó, se envía un mensaje de error a la interfaz informando el error.

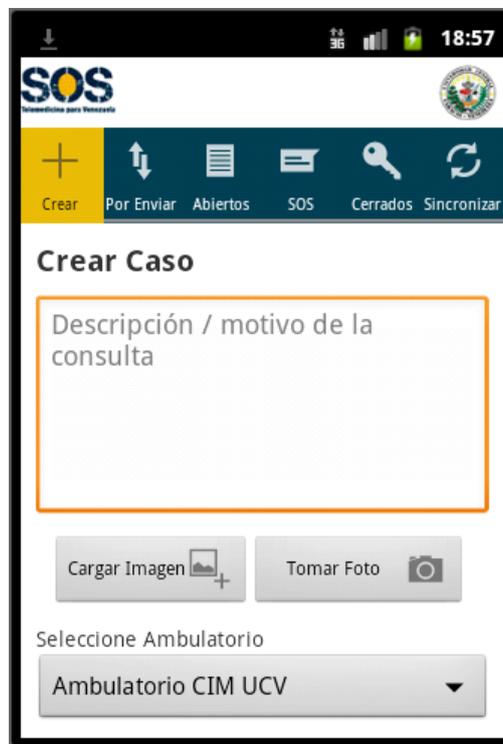


Figura 57. Interfaz de Usuario Medico Rural

- **Crear Caso**

Esta opción permite al médico rural registrar un nuevo caso para que pueda ser tramitado y obtener la opinión de especialistas.

### **Interfaz de Usuario**

Desde el punto de vista de la interfaz de usuario se diseñó la pantalla de creación de casos que le permita al médico rural crear un nuevo caso. Para ello tenemos los siguientes campos: “Descripción/Motivo de la consulta”, “Cargar Imagen”, “Tomar Foto”, “Seleccione Ambulatorio”, y un botón “Crear Caso”.

En el campo de texto “Descripción/Motivo de la consulta” se implementó un campo de tipo “EditText” que permite que ese campo sea tan extenso como se necesite redimensionándose así al tamaño de la pantalla haciendo más cómodo el ingreso de datos desde cualquier dispositivo móvil. Al seleccionar este campo se despliega el teclado por defecto del dispositivo móvil para el ingreso de los datos como se puede apreciar en la figura 58.

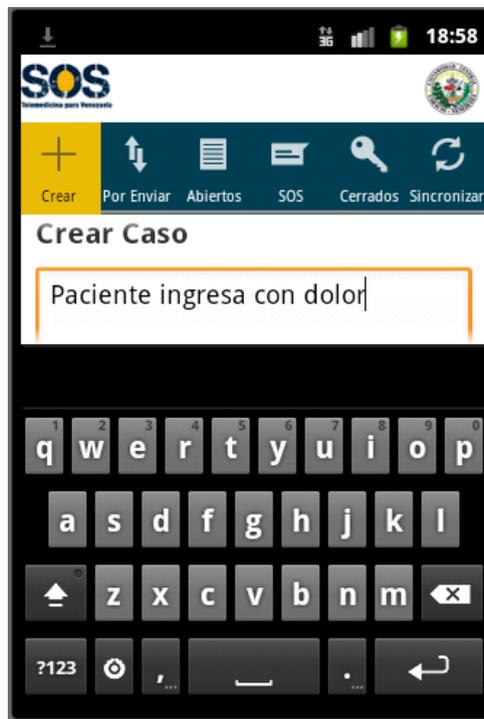


Figura 58. Descripción de Pantalla Crear Caso

Para el campo “Centro Asistencial” se realizó una lista desplegable que contiene los centros ambulatorios disponibles.

Para el campo “Cargar imagen” se implementó la posibilidad de cargar un archivo desde la galería del dispositivo móvil y adjuntarlo a la aplicación. Para ello se hace un llamado desde el botón `subir_imagen` a una nuevo “intent” para crear un chooser o seleccionador que nos permita abrir la galería y seleccionar un elemento. Ver figura 59.

```
case R.id.subir_imagen:  
    Intent intent = new Intent();  
    intent.setType("image/*");  
    intent.setAction(Intent.ACTION_GET_CONTENT);  
    startActivityForResult(  
        Intent.createChooser(intent, "Select Picture"),  
        SELECT_PICTURE);  
break;
```

Figura 59. Código para cargar imagen al servidor Android

Luego se valida el resultado de este “intent” y si el resultado es OK, es decir si se seleccionó alguna imagen, se guarda el directorio de la imagen y se le asigna al `ImageView` de la interfaz para mostrar una vista previa de la imagen seleccionada

En la figura 60, se muestra la vista del *chooser* o seleccionador de la galería desde la interfaz de usuario



Figura 60. Seleccionador de Imagen de La Galería

Para el campo “Tomar Foto” se implementó la funcionalidad de desplegar la cámara del dispositivo para capturar una foto en tiempo real y cargarla en la aplicación. Para ello se hace un llamado desde el botón tomar\_foto a la función captureImage() que activará un nuevo intent que desplegará la cámara y permitirá tomar una fotografía y almacenarla en el dispositivo. Ver figura 61.

```
private void captureImage() {  
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
  
    fileUri = getOutputMediaFileUri(MEDIA_TYPE_IMAGE);  
  
    intent.putExtra(MediaStore.EXTRA_OUTPUT, fileUri);  
  
    // start the image capture Intent  
    startActivityForResult(intent, CAMERA_CAPTURE_IMAGE_REQUEST_CODE);  
}
```

Figura 61. Código para capturar imagen desde el dispositivo. Android

Esto desplegará la cámara del dispositivo y permitirá capturar una fotografía.

Una vez realizado, se muestra la imagen capturada desde la cámara en el ImageView de la interfaz de usuario invocando a la función previewCapturedImage() dentro de la cual se va a codificar la imagen capturada en un Bitmap o mapa de bits para luego poder asignarle este Bitmap al campo de imagen image1 de la interfaz de usuario

En ambos casos, la imagen seleccionada desde la galería o la imagen tomada desde la cámara del dispositivo se muestra en la pantalla de la aplicación como una vista previa tal y como se puede apreciar en la figura 62.

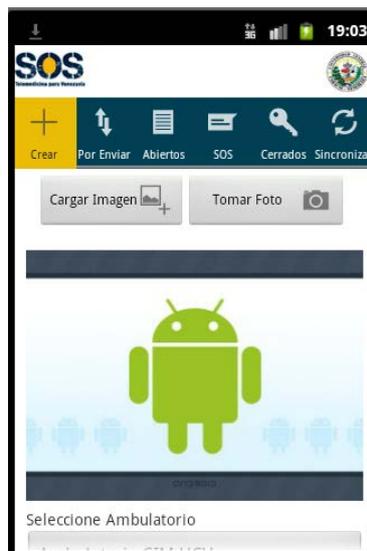


Figura 62. Pantalla Vista Previa de Imagen

Luego de tener los datos del caso, el ambulatorio seleccionado y la imagen cargada (opcional) se presiona el botón “Crear Caso” y se realiza la petición para almacenar el caso dentro de la base de datos local SQLite.

### **Almacenando el Caso**

Una vez seleccionado el botón “Crear Caso” se instancia un objeto de la clase Caso() y se le asignan los atributos, los cuales se obtienen desde la interfaz del usuario. Luego de esto se llama a la clase DataSource() específicamente a la función dataSource.crearCaso(caso); que se encargará de insertar el objeto del tipo Caso() en la tabla casos de la base de datos Rural.sql de la aplicación.

Si la interfaz de usuario contiene una imagen, se crea un objeto de tipo ArchivoUp() para poder almacenar el archivo en la tabla archivoup de nuestra base de datos local Rural.sql. Para ello se llama a la función dataSource.subirArchivo() a la que se le pasa el objeto del tipo ArchivoUp() y la URL de la imagen que se almacenará.

Una vez almacenado el caso y la imagen, se devuelve el control a la interfaz gráfica del usuario

- **Casos Por Enviar (Pendientes)**

Permite consultar el histórico de casos que ha generado el médico rural registrado pero que no han sido subidos al servidor, es decir no están en la base de datos de SOS-Telemedicina sino que por el contrario están solamente almacenados de forma local en el dispositivo. Cada caso muestra a su vez detalles como su descripción, estado actual del caso, ambulatorio seleccionado, fecha de creación y archivos que contenga el mencionado caso y que no han sido enviados al servidor.

### **Interfaz**

En la interfaz “Pendientes”, ver figura 63, se implementó una lista desplegable que permite ver un resumen del contenido del caso pendiente y cuando la seleccionamos se expande mostrando todo el detalle del caso a continuación. Esto permite incluir gran cantidad de ítems en forma de lista desplegable en una misma pantalla ideal para dispositivos móviles sin necesidad de interactuar con diferentes pantallas para obtener la información.



Figura 63. Pantalla Casos Por Enviar

Adicionalmente se implementó un campo de notificación que indicará cuando el médico tiene casos pendientes por enviar mostrándolo de color rojo y dándole la posibilidad de presionar el botón de sincronizar para enviar estos casos con sus respectivos archivos al servidor. Si el usuario Médico Rural no tiene casos pendientes por enviar, el campo en cuestión y el botón desaparecerán.

Para la implementación de la lista desplegable se utilizó un adaptador o “Adapter” personalizado llamado `CasosPorEnviarAdapter.java` que manipula objetos de la clase `Caso()`, lo que permite incluir objetos del tipo `Caso()` en esta lista.

A esta lista se le creó la posibilidad de que sea seleccionable por el usuario y redireccione al usuario a otra pantalla con el detalle del caso que se seleccionó. Para ello la lista implementa la clase `OnItemClickListener()` específicamente el método `setOnItemClickListener()` donde al seleccionar un ítem de la lista, se hace una redirección al caso específico que el usuario seleccionó.

Dentro del detalle del caso, se puede apreciar diferentes campos a mostrar, entre los que destacamos el link a Ver Archivos el cual se implementa de la siguiente manera:

- Archivos: Campo seleccionable que permite tener acceso a los archivos almacenados en la base de datos. Al seleccionar el archivo, se realiza una petición a la base de datos a la tabla archivosup y devuelve la imagen que se tomó asociada al caso.

### **Obteniendo datos de la Base de Datos**

Una vez seleccionado el caso de la lista desplegable de casos, se muestra una pestaña con el detalle de los casos asociados al caso solicitado. Para ello se realizó una consulta a la bases de datos local para obtener todos los detalles del caso según el ID seleccionado por el usuario invocando a la clase DataSource.java mas específicamente a la función dataSource.getCasoById(); el cual retorna un objeto del tipo MisCasos() con los datos del caso que se desea consultar.

Una vez retornado el caso en cuestión, se asignan los valores a la interfaz de usuario para que se muestre la información por pantalla.

Luego hay que validar si el caso tiene asociado archivos adjuntos para poder mostrarlos en los detalles del caso.

Si el usuario no tiene archivos asociados la interfaz de usuario no mostrará dicha información, sin embargo, si por el contrario si posee archivos, se mostrará en la interfaz el campo seleccionable. Para ello se implementó un condicional que valida la existencia de archivos relacionados al caso

- **Abiertos**

Esta funcionalidad le permite al usuario consultar el histórico de casos que ha generado el médico rural registrado y que aún no han sido resueltos por un especialista, es decir que se encuentran abiertos. Cada caso muestra a su vez detalles como su descripción, estado actual del caso, tiempo que tiene en espera, datos del paciente que tiene asociado dicho caso, opiniones y archivos que contenga el mencionado caso y que ya fueron enviados al servidor.

## Interfaz

En la interfaz “Abiertos”, ver figura 64, se implementó una lista desplegable que permite ver un resumen del contenido y cuando la seleccionamos se expande mostrando todo el detalle del caso a continuación. Esto permite incluir gran cantidad de ítems en forma de lista desplegable en una misma pantalla ideal para dispositivos móviles sin necesidad de interactuar con diferentes pantallas para obtener la información.



Figura 64. Pantalla de Casos Abiertos

Para la implementación de la lista desplegable se utilizó un adaptador o “Adapter” personalizado llamado `MisCasosAdapter.java` que manipula objetos de la clase `MisCasos()`, lo que permite incluir objetos del tipo `Miscasos()` en esta lista.

A esta lista se le creó la posibilidad de que sea seleccionable por el usuario y redireccione al usuario a otra pantalla con el detalle del caso que se seleccionó. Para ello la lista implementa la clase `OnItemClickListener()` específicamente el método `setOnItemClickListener()` donde al seleccionar un ítem de la lista, se hace una redirección al caso específico que el usuario seleccionó.

Dentro del detalle del caso, ver figura 64, se puede apreciar diferentes campos a mostrar, entre los que destacamos los enlaces a “Ver Opiniones” y “Ver Archivos”, los cuales se implementaron de la siguiente manera:

- **Opiniones:** campo seleccionable que permite observar las opiniones asociadas al caso. Al seleccionar la opinión se realiza una petición a la base de datos local a la tabla respuesta que devuelve el detalle de la opinión solicitada.
- **Archivos:** campo seleccionable que permite tener acceso a los archivos almacenados en la base de datos. Al seleccionar el archivo, se realiza una petición a la base de datos a la tabla archivos y devuelve la imagen que se tomó asociada al caso.
- **Cerrar Caso:** esta funcionalidad permite a un médico rural cerrar el caso que está consultando. Para ello debe presionar el botón “Cerrar” disponible en el detalle del caso para enviar la solicitud al servidor. Del lado del servidor al momento de cerrar un caso se cambia el estatus del mismo en la Base de datos a estatus “Cerrado” y se actualizan tanto la tabla casos para modificar el estatus, así como la tabla historial para actualizar el historial de ejecución que nos permitirá llevar un control de todas las transacciones realizadas sobre ese caso en particular y sobre todo el sistema. Ver figura 65.



Figura 65. Pantalla Detalle de Casos

## **Obteniendo datos de la Base de Datos**

Una vez seleccionado el caso de la Lista desplegable de casos, se muestra una pestaña con el detalle correspondiente al caso seleccionado. Para ello se realizó una consulta a la bases de datos local para obtener todos los detalles del caso según el ID seleccionado por el usuario invocando a la clase DataSource.java mas específicamente a la función dataSource.getCasoById(); el cual retorna un objeto del tipo MisCasos() con los datos del caso que se desea consultar.

Una vez retornado el caso en cuestión, se asignan los valores a la interfaz de usuario para que se muestre la información por pantalla. Luego hay que validar si el caso tiene asociado opiniones o archivos adjuntos para poder mostrarlos en los detalles del caso.

Si el usuario no tiene archivos asociados u opiniones asociadas la interfaz de usuario no mostrará dicha información, sin embargo, si por el contrario si posee opiniones o archivos, se mostrará en la interfaz el campo seleccionable. Para ello se implementó un condicional que valida la existencia de opiniones o archivos relacionados al caso

- **SOS – Casos Resueltos**

Esta funcionalidad le permite al Médico Rural consultar los casos que están en el sistema que tiene estatus “Resuelto”, es decir que fueron respondidos por un médico especialista. Cada caso muestra a su vez detalles como su descripción, estado actual del caso, tiempo que tiene en espera, datos del paciente que tiene asociado dicho caso, opiniones y archivos que contenga el mencionado caso y que ya fueron enviados al servidor.

La implementación de esta funcionalidad es similar a la funcionalidad “Abierto” explicada previamente.

- **Cerrados**

Esta funcionalidad le permite al Médico Rural consultar los casos que están en el sistema que tiene estatus “Cerrados”, es decir que fueron cerrados por el médico rural. Cada caso muestra a su vez detalles como su descripción, estado actual del caso, tiempo que tiene en espera, datos del paciente que tiene asociado dicho caso, opiniones y archivos que contenga el mencionado caso y que ya fueron enviados al servidor.

La implementación de esta funcionalidad es similar a la funcionalidad “Abierto” explicada previamente.

- **Sincronizar**

Se encarga de sincronizar la base de datos local de la aplicación con la base de datos del servidor de SOS-TRIAJE.

Cuando el médico rural está en una zona geográfica en donde posea cobertura y posea conexión a Internet, bien sea por conexión de datos o por Wi-fi, puede presionar el botón “Sincronizar” para comenzar la sincronización de ambas bases de datos (Dispositivo-Servidor). Ver figura 66. En caso de que presione el botón en una zona sin cobertura, el sistema devolverá un mensaje de error indicando que no es posible realizar la conexión.

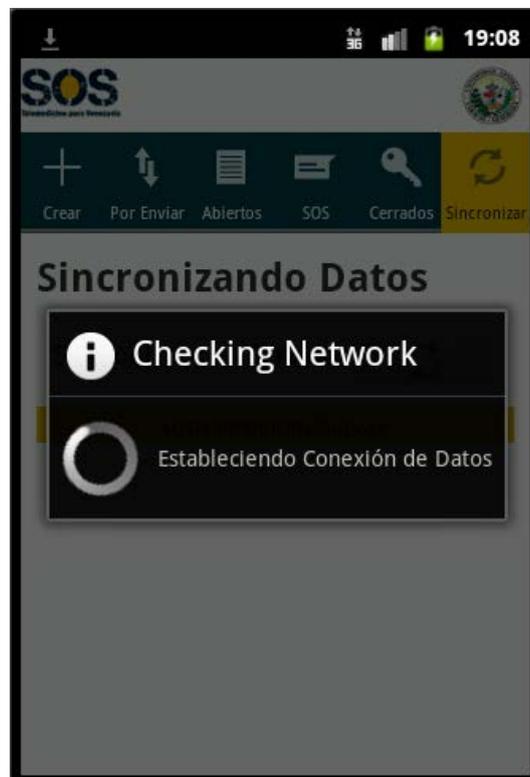


Figura 66. Pantalla Sincronizar Datos Con El Servidor SOS

Para realizar la sincronización del dispositivo con el servidor se implementaron seis hilos de ejecución que se detallan a continuación:

- **Hilo de ejecución para validar conexión de datos a Internet.(  
NetCheck.java)**

Desde la clase Sincronizar.java se llama al primer hilo de ejecución “NetCheck.java” que se ejecutará de manera asíncrona en background y cuya función principal es validar que exista una conexión a Internet haciendo una prueba de conectividad hacia una página común en internet, en nuestro caso a www.google.com.

Si la respuesta de esa conexión es positiva, se hace el llamado al siguiente hilo de ejecución. En caso de no tener conexión con Internet se manda a la interfaz un mensaje de error de conexión y retornamos a la pantalla principal de la aplicación.

- **Hilo de ejecución para subir los casos almacenados en la base de datos local hacia la base de datos del servidor (AsyncUpCasos.java).**

Una vez verificada la conexión con Internet se crea una Lista de objetos del tipo Caso() que contendrá todos los casos almacenados en la base de datos local del dispositivo móvil específicamente en la tabla caso para poder subirlos al servidor.

Luego que es obtenida la lista de casos desde la base de datos local, se hace un llamado a la clase asíncrona AsyncUpCasos.java la cual se encargará de parsear cada caso contenido en la lista de casos a formato String para ser enviado vía JSON al servidor. Una vez enviado al servidor retornará un contador con la cantidad de casos totales existentes en la base de datos del servidor.

- **Hilo de ejecución para subir archivos almacenados en la base de datos local hacia la base de datos del servidor (AsyncSubirArchivos.java).**

Seguidamente de haber subido los casos al servidor, se crea una Lista de objetos del tipo ArchivoUp() que contendrá todos los archivos almacenados en la base de datos local del dispositivo móvil específicamente en la tabla archivoup para que sean subidos al servidor. Para esto se hace un llamado a la clase Datasource.java específicamente a la función dataSource.getAllArchivoUp();

Luego que es obtenida la lista de archivos desde la base de datos local, se hace un llamado a la clase asíncrona `AsyncSubirArchivos.java` la cual se encargará de parsear cada archivo contenido en la lista de archivosup a formato String para ser enviado vía JSON al servidor.

En esta función hay un reto adicional el cual es transformar una imagen almacenada en la tabla archivo a String para ser enviada vía JSON al servidor. Para esto utilizamos la librería `android.util.Base64`, específicamente a la función `Base64.encodeToString()` que nos permitirá parsear una imagen almacenada en formato BLOB en una base de datos a una cadena de caracteres formato String.

Una vez enviado la lista de archivos al servidor se hace un llamado a la clase `Datasource.java` específicamente a la función `dataSource.LimpiarTablasDownload()`; que simplemente eliminará el contenido de las tablas `miscasos`, `archivo` y `respuestas` para poder descargar los casos que están en el servidor y almacenarlos en la base de datos del dispositivo

- **Hilo de ejecución para descargar los casos almacenados en la base de datos del servidor hacia la base de datos local en el dispositivo. (`AsyncDownMisCasos.java`)**

Luego de haber limpiado las tablas para realizar las descargas desde el servidor hacia el dispositivo, se crea una Lista de objetos del tipo `MisCasos ()` que contendrá todos los casos almacenados en el servidor asociados al médico rural autenticado, y que se pasarán a la base de datos local del dispositivo móvil específicamente en la tabla `miscasos`.

Para obtener esta lista desde el servidor se hace un llamado a la clase asíncrona `AsyncDownMisCasos.java` la cual se encargará de parsear cada caso obtenido desde el servidor en formato JSON a un objeto del tipo `MisCasos()`.

Una vez obtenida la lista de `MisCasos()` provenientes del servidor, se recorre la lista y se van insertando uno por uno los objetos en la base de datos local del dispositivo móvil en la tabla `miscasos`.

- **Hilo de ejecución para descargar las respuestas almacenadas en la base de datos del servidor hacia la base de datos local en el dispositivo. (`AsyncDownRespuestas.java`)**

Luego de haber descargados los casos hacia el dispositivo móvil, se crea una Lista de objetos del tipo Respuestas() que contendrá todas las respuestas almacenadas en el servidor asociadas al médico rural autenticado, y que se pasarán a la base de datos local del dispositivo móvil específicamente en la tabla respuestas.

Para obtener esta lista desde el servidor se hace un llamado a la clase asíncrona AsyncDownRespuestas.java la cual se encargará de parsear cada respuesta obtenida desde el servidor en formato JSON a un objeto del tipo Respuestas().

Una vez obtenida la lista de Respuestas () provenientes del servidor, se recorre la lista y se van insertando uno por uno los objetos en la base de datos local del dispositivo móvil en la tabla respuestas

- **Hilo de ejecución para descargar los archivos almacenados en la base de datos del servidor hacia la base de datos local en el dispositivo.( AsyncDownArchivos.java)**

Luego de haber descargados las respuestas hacia el dispositivo móvil, se crea una Lista de objetos del tipo Archivo() que contendrá todos los archivos almacenados en el servidor asociados a los casos del médico rural autenticado, y que se pasarán a la base de datos local del dispositivo móvil específicamente en la tabla archivo.

Para obtener esta lista desde el servidor se hace un llamado a la clase asíncrona AsyncDownArchivos.java la cual se encargará de parsear cada archivo obtenido desde el servidor en formato JSON a un objeto del tipo Archivo().

En esta función debemos transformar una imagen proveniente del servidor en formato JSON para ser almacenada en nuestra base de datos en formato BLOB. Para esto utilizamos la librería android.util.Base64, específicamente la función Base64.decode() que nos permitirá parsear una imagen en formato String en un arreglo de Bytes para que luego pueda ser almacenada en una base de datos en un campo de tipo BLOB.

Una vez obtenida la lista de Archivo() provenientes del servidor, se recorre la lista y se van insertando uno por uno los objetos en la base de datos local del dispositivo móvil en la tabla archivos

Una vez descargada la lista de archivos a la base de datos local, se hace un llamado a la clase Datasource.java específicamente a la función dataSource.LimpiarTablasUpload(); que simplemente eliminará el contenido de las tablas caso y archivoup para poder almacenar los nuevos casos y archivos que se creen en el dispositivo móvil.

- **Salir del Sistema**

Salir de forma controlada del sistema cerrando la sesión del usuario actual.

Se puede salir del sistema desde el menú del dispositivo móvil seleccionando la opción “Cerrar Sesión”. En la figura 67 se puede apreciar la interfaz de la aplicación con el botón “Cerrar Sesión”,



Figura 67. Pantalla Cerrar Sesión

Al presionar este botón, se envía una petición a la función “cerrar sesión” del menú de la aplicación. Dentro de esta acción se eliminan las variables de preferencia o sesión y cualquier

dato que se había almacenado en la variable SharedPreferences que como vimos anteriormente, nos permite almacenar datos o preferencias para ser compartidos entre otros métodos o actividades dentro de nuestra aplicación Android de forma persistente.

Luego de esto cierra cualquier conexión remanente con la base de datos y finalmente redirecciona a la clase Login.java para una nueva autenticación.

#### 4.5.4. Etapa IV. Pruebas

A continuación se muestran el conjunto de pruebas funcionales de tipo caja cerrada que se realizaron. Se destaca que generalmente se recomienda que el equipo de desarrollo no sea el mismo que realice las pruebas sobre las funcionalidades del sistema, es por eso que en las actividades de pruebas se contó con la participación del equipo del Centro de Informática Médica.

A continuación se describen las pruebas realizadas en un total de 6 Tablas.

**Tabla 22. Prueba de funcionamiento 1. Android**

|                                 |  | PRUEBAS        |          |
|---------------------------------|--|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Inicio de sesión</b>  | <b>Prueba:</b> | <b>1</b> |
| <b>Objetivo:</b>                | Constatar el inicio de sesión del usuario médico de triaje, médico especialista y médico rural.                          |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>Se ingresa para el rol del sistema, el nombre de usuario y contraseña.</li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria  |                |          |
| <b>Resultado Obtenido:</b>      | El usuario accede al sistema y puede realizar acciones según su perfil.  |                |          |
| <b>Observaciones:</b>           | Se constata que efectivamente el usuario ingresa al sistema de manera correcta.  |                |          |

Tabla 23. Prueba de funcionamiento 2. Android

|                                 |  | PRUEBAS        |          |
|---------------------------------|--|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Creación de un Nuevo Caso</b>   | <b>Prueba:</b> | <b>2</b> |
| <b>Objetivo:</b>                | Registrar un nuevo caso en la base de datos local del dispositivo  |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se ingresan los datos para la creación del caso               <ul style="list-style-type: none"> <li>○ Descripción del caso</li> <li>○ Centro Asistencial</li> <li>○ Archivo Adjunto (Opcional)</li> </ul> </li> <li>• Se pasa el Id del médico y la fecha actual mediante una consulta interna.</li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria  |                |          |
| <b>Resultado Obtenido:</b>      | Visualización del Caso Creado en la base de datos local del dispositivo con los datos que coinciden con los ingresados   |                |          |
| <b>Observaciones:</b>           | Se pudo constatar el correcto funcionamiento de esta actividad sin necesidad de tener conexión a datos o internet.   |                |          |

Tabla 24. Prueba de funcionamiento 3. Android

|                                 |  | PRUEBAS        |          |
|---------------------------------|--|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Consulta de Casos - Médico Rural</b>  | <b>Prueba:</b> | <b>3</b> |
| <b>Objetivo:</b>                | Ver el detalle de los casos registrados en el sistema según el estatus seleccionado desde el perfil Médico Rural.  |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona el tipo de caso a ser consultado               <ul style="list-style-type: none"> <li>○ Casos Pendientes</li> <li>○ Casos Resueltos</li> <li>○ Casos Cerrados</li> <li>○ Se pasa el Id del médico y la fecha actual mediante una consulta interna.</li> </ul> </li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria  |                |          |
| <b>Resultado Obtenido:</b>      | Visualización del detalle del caso seleccionado según su estatus en sistema.   |                |          |
| <b>Observaciones:</b>           |  |                |          |

Tabla 25. Prueba de funcionamiento 4. Android

|                                 |  | PRUEBAS        |          |
|---------------------------------|--|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Consulta de Casos Por Enviar</b>  | <b>Prueba:</b> | <b>4</b> |
| <b>Objetivo:</b>                | Ver todos los casos pendientes por enviar al servidor.   |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>Se selecciona la pestaña “Por Enviar”</li> <li>Se selecciona el (los) registro (s) a consultar para evaluar el detalle del caso.</li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria  |                |          |
| <b>Resultado Obtenido:</b>      | Visualización del resumen del caso seleccionado, que está pendiente por enviarse al servidor.  |                |          |
| <b>Observaciones:</b>           | Se pudo constatar el correcto funcionamiento de esta actividad sin necesidad de tener conexión a datos o internet.   |                |          |

Tabla 26. Prueba de funcionamiento 5. Android

|                                 |   | PRUEBAS        |          |
|---------------------------------|---|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Cargar Imagen desde Galería</b>  | <b>Prueba:</b> | <b>5</b> |
| <b>Objetivo:</b>                | Cargar una imagen desde la galería del dispositivo y almacenarla en la base de datos del sistema.   |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>Se selecciona el botón “Cargar Imagen”</li> <li>Se selecciona desde la galería del dispositivo, la imagen que se desea adjuntar.</li> <li>El sistema valida el tamaño y el formato de la imagen.</li> <li>Se carga una vista previa de la imagen en la interfaz del usuario.</li> </ul>        |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria.  |                |          |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>Se carga la vista previa de la imagen en la interfaz del usuario.</li> <li>Se almacena la imagen en la base de datos local del dispositivo para luego ser enviada al servidor.</li> <li>Se almacena la imagen en la base de datos del sistema cuando se sincroniza con el servidor.</li> </ul> |                |          |
| <b>Observaciones:</b>           | Se presentaron inconvenientes al momento de enviar las imágenes desde la base de datos local al servidor. Se resolvieron los problemas de forma satisfactoria implementando el “parseo” de datos utilizando la tecnología JSON.   |                |          |

Tabla 27. Prueba de funcionamiento 6. Android

|                                 |   | PRUEBAS        |          |
|---------------------------------|---|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Cargar Imagen desde la Cámara del dispositivo</b>  | <b>Prueba:</b> | <b>6</b> |
| <b>Objetivo:</b>                | Activar la cámara del dispositivo, capturar una imagen y almacenarla en la base de datos del sistema.   |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona el botón “Tomar Foto”</li> <li>• Se activa la cámara y se captura la imagen que se desea adjuntar al caso.</li> <li>• El sistema valida el tamaño y el formato de la imagen capturada.</li> <li>• Se carga una vista previa de la imagen en la interfaz del usuario.</li> </ul>      |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria.  |                |          |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>• Se carga la vista previa de la imagen en la interfaz del usuario.</li> <li>• Se almacena la imagen en la base de datos local del dispositivo para luego ser enviada al servidor.</li> <li>• Se almacena la imagen en la base de datos del sistema cuando se sincroniza con el servidor.</li> </ul> |                |          |
| <b>Observaciones:</b>           | Se presentaron inconvenientes al momento de activar la cámara del dispositivo. Se resolvieron los problemas de forma satisfactoria.   |                |          |

Tabla 28. Prueba de funcionamiento 7. Android

|                                 |  | PRUEBAS        |          |
|---------------------------------|--|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Cierre de un Caso</b>   | <b>Prueba:</b> | <b>7</b> |
| <b>Objetivo:</b>                | Cerrar un caso creado previamente por el médico rural.   |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona el caso a ser cerrado.</li> <li>• Se selecciona el botón “Cerrar”</li> <li>• Se pasa el Id del médico y la fecha actual mediante una consulta interna.</li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria.   |                |          |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>• El caso queda cerrado por el médico rural asociado y su estatus es “Cerrado”</li> <li>• Visualización del caso cerrado asociado al médico especialista.</li> </ul>              |                |          |
| <b>Observaciones:</b>           | Esta actividad requiere conexión a internet o a datos.   |                |          |

Tabla 29. Prueba de funcionamiento 8. Android

|                                 |  | PRUEBAS        |          |
|---------------------------------|--|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Ver Imagen Adjunta</b>  | <b>Prueba:</b> | <b>8</b> |
| <b>Objetivo:</b>                | Visualizar todas las imágenes que tiene asociado un caso.  |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona el caso a consultar.</li> <li>• Se selecciona el nombre de la imagen que se desea consultar.</li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria.   |                |          |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>• Se despliega en pantalla la imagen seleccionada asociada al caso</li> </ul>   |                |          |
| <b>Observaciones:</b>           |  |                |          |

Tabla 30. Prueba de funcionamiento 9. Android

|                                 |   | PRUEBAS        |          |
|---------------------------------|---|----------------|----------|
| <b>Nombre de la prueba:</b>     | <b>Ver Opiniones del Caso</b>   | <b>Prueba:</b> | <b>9</b> |
| <b>Objetivo:</b>                | Visualizar todas las opiniones asociadas a un caso.   |                |          |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>• Se selecciona el caso a consultar.</li> <li>• Se selecciona el nombre de la opinión que se desea consultar.</li> </ul> |                |          |
| <b>Evaluación de la prueba:</b> | Satisfactoria.  |                |          |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>• Se despliega en pantalla la opinión seleccionada asociada al caso.</li> </ul>  |                |          |
| <b>Observaciones:</b>           |   |                |          |

**Tabla 31. Prueba de funcionamiento 10. Android**

|                                 |  | PRUEBAS        |           |
|---------------------------------|--|----------------|-----------|
| <b>Nombre de la prueba:</b>     | <b>Sincronizar Datos.</b>  | <b>Prueba:</b> | <b>10</b> |
| <b>Descripción:</b>             | Sincronizar los datos almacenados en la base de datos local del dispositivo con el servidor.   |                |           |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>Se selecciona la pestaña “Sincronizar”</li> </ul>   |                |           |
| <b>Evaluación de la prueba:</b> | Satisfactoria.   |                |           |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>Se almacenan en la base de datos del servidor los casos y archivos almacenados previamente de manera local en la base de datos local del dispositivo.</li> <li>Se almacenan en el dispositivo, los casos y archivos que están en el servidor pero que pertenecen al médico rural autenticado para que sean accesibles de manera local.</li> </ul> |                |           |
| <b>Observaciones:</b>           | Esta actividad requiere conexión a internet o a datos.   |                |           |

**Tabla 32. Prueba de funcionamiento 11. Android**

|                                 |   | PRUEBAS        |           |
|---------------------------------|---|----------------|-----------|
| <b>Nombre de la prueba:</b>     | <b>Salir del Sistema</b>  | <b>Prueba:</b> | <b>11</b> |
| <b>Objetivo:</b>                | Salir del sistema de forma controlada   |                |           |
| <b>Entradas:</b>                | <ul style="list-style-type: none"> <li>Se selecciona el botón “Cerrar Sesión”</li> </ul>  |                |           |
| <b>Evaluación de la prueba:</b> | Satisfactoria.  |                |           |
| <b>Resultado Obtenido:</b>      | <ul style="list-style-type: none"> <li>Se cierra sesión del usuario borrando los datos de sesión almacenados.</li> <li>Se visualiza la pantalla de inicio indicando que se ha desconectado de forma exitosa.</li> </ul> |                |           |
| <b>Observaciones:</b>           |   |                |           |

## **CAPÍTULO V – RESULTADOS GENERALES**

A partir de los objetivos planteados se obtuvieron los siguientes resultados:

Con la finalidad de Analizar en detalle el proceso asistencial llevado a cabo en los centros de salud integrantes del proyecto SOS Telemedicina, se realizó un levantamiento de información y un estudio de los proceso llevado a cabo en los centros asistenciales con respecto al manejo de historia clínica, al procedimiento de referencias médicas y los beneficios que presenta la utilización de los sistemas desarrollados dentro del Programa SOS Telemedicina para Venezuela. Se detectaron las necesidades actuales de comunicación presentadas en dichos centros de salud, además se identificaron los requerimientos por parte de los profesionales de la salud, así como el beneficio del diseño y desarrollo de un sistema móvil utilizable y aplicable en este entorno. Este análisis permitió diseñar una forma de emular ese proceso para trabajar en entornos móviles, de modo que permita la gestión electrónica de segunda opinión a través de dispositivos móviles tomando en cuenta las soluciones existentes y los requerimientos específicos identificados.

Mediante la información recolectada y analizada, además del estudio de sistemas actuales de referencias médicas, se pudo modelar el flujo de comunicación que actualmente utilizan los centros asistenciales para la obtención de una segunda opinión en salud. A partir de esto se plantearon funcionalidades que dieran valor agregado a este proceso y que permitieran así mejorarlo y complementarlo bajo las tecnologías de desarrollo móvil.

El Sistema de Referencias Médicas (SOS-TRIAJE), ofrece beneficios en el proceso de emisión de opiniones entre profesionales de la salud en un determinado caso. A través del estudio del mismo, se pudieron conocer a fondo las herramientas tecnológicas y protocolos de comunicación para la transferencia de datos clínicos de forma estandarizada implementados, para lograr así la comunicación con el Sistema de Historia Médica Electrónica (SOS-HME) y poder lograr de una manera más rápida el proceso actual de segunda opinión en salud. Herramientas de software en el ámbito web, base de datos propia del sistema, comunicación a través de servicios, entre otras, fueron las bases para el desarrollo del sistema móvil.

Por su parte, el estudio de los sistemas y herramientas de software empleadas actualmente en el ámbito de salud en dispositivos móviles, permitieron determinar el diseño del sistema móvil realizado, conocer así a profundidad la utilización de los dispositivos móviles por parte de

profesionales de la salud, en la solución de casos, consultas, atención, tratamientos, y demás funciones que actualmente ofrece de manera eficiente la comunicación móvil, para poder estructurar un sistema que basado en la experiencia de usuarios permita ofrecer ventajas y nuevas oportunidades a los centros asistenciales en lugares remotos de nuestro país.

Teniendo como base el sistema previamente desarrollado para el manejo de segundas opiniones y triaje ambulatorio (SOS-TRIAJE), se pudo implementar una aplicación web móvil que funciona en cualquier dispositivo mediante un navegador web y con acceso a Internet, ofreciendo el conjunto de funcionalidades actuales que posee cada usuario y llevándolas al ámbito móvil, además de permitir que el nuevo sistema muestre no sólo las funciones de consulta de datos, envío de patologías, consulta de respuestas, evolución de un caso, sino que además contempla un nuevo usuario (Médico Rural), que formando parte del sistema puede crear nuevos casos y así complementar el proceso de segunda opinión en salud en centros asistenciales.

Así mismo, se llevó a cabo el desarrollo de una aplicación nativa en Android, la cuál le permite al usuario Médico Rural desde un dispositivo móvil realizar las funcionalidades propias de su perfil y además crear nuevos casos sin la limitación de estar conectado a Internet, y poder enviar dichos casos cuando se encuentre en un ambiente con conectividad. Esta funcionalidad agrega valor al sistema móvil y ofrece una funcionalidad más dentro del programa SOS Telemedicina para Venezuela.

Finalmente, se realizaron pruebas funcionales luego del desarrollo de cada sistema móvil. Las mismas resultaron satisfactorias y permitieron la corrección y mejoramiento de varias funcionalidades. Las pruebas se basaron en las acciones que puede realizar cada usuario dentro del sistema, y en analizar el flujo dentro del mismo.

Una vez analizados los requerimientos técnicos necesarios y desarrollados los dos aplicativos, se procedió a integrar el Sistema Móvil de Referencias Médicas con el Sistema SOS-TRIAJE existente. Para ello, se realizó la implementación de servicios web que permiten enviar y recibir información entre los sistemas móviles y la base de datos del sistema actual alojada en el servidor. La arquitectura resultante de ésta integración se puede apreciar en la figura 68.

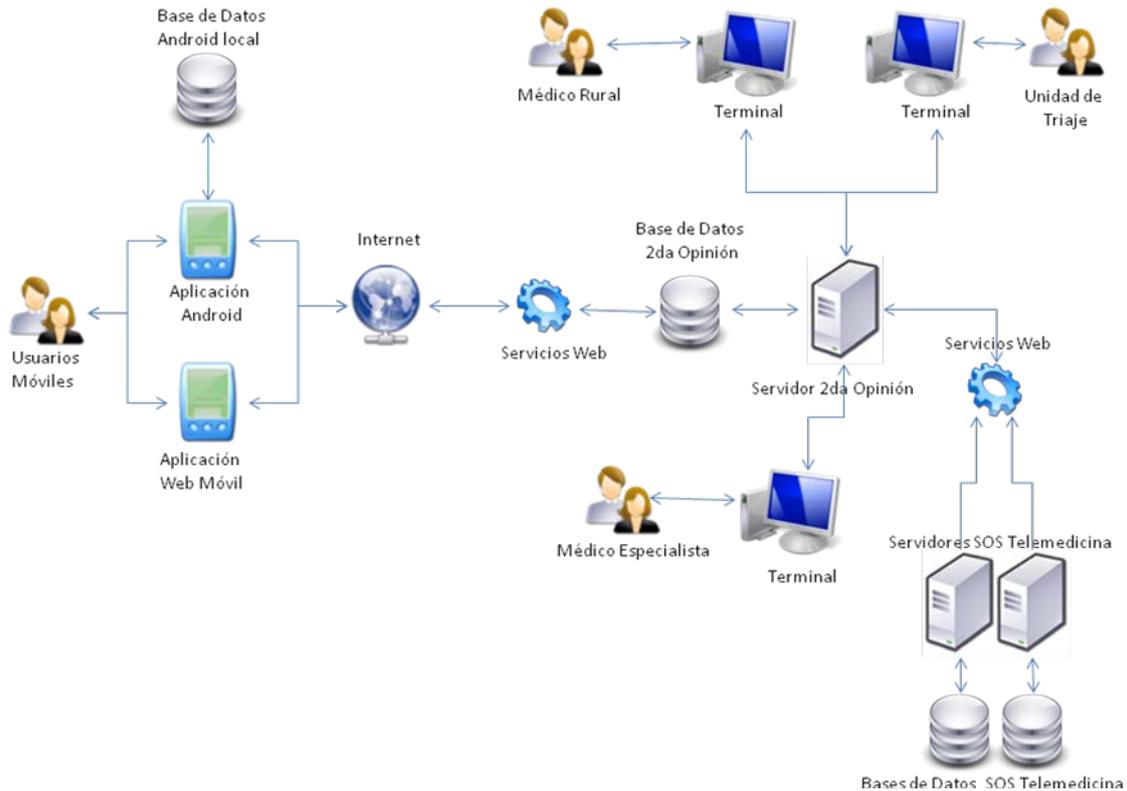


Figura 68. Arquitectura General de integración entre Sistemas

Aprovechando las bondades de las bases de datos SQLite para el sistema operativo Android, se manejó la persistencia de datos de manera local en la aplicación móvil Nativa Android permitiendo que el sistema funcione sin necesidad de tener conexión a Internet.

Para garantizar la consistencia de datos entre el dispositivo móvil y la base de datos del servidor, se manejan versiones de las tablas de la base de datos y se desarrolló un proceso de sincronización controlado que permite enviar y recibir datos entre el servidor y el dispositivo Android de forma segura, ágil y controlada.

Como resultado de la utilización del patrón de diseño MVC, se obtuvo un Sistema de Móvil de Referencias Médicas modular, con clara separación de la interfaz, datos y la lógica del negocio lo que le permite que sea mantenible, reutilizable, y de fácil modificación en el tiempo.

Se implementaron mecanismos para garantizar la seguridad en el sistema teniendo como requisito principal la autenticación de los usuarios validando los datos ingresados con los datos almacenados en la base de datos. Adicionalmente se aplicaron mecanismos de cifrado para almacenar las contraseñas en la base de datos y en el sistema. El envío de la información entre las aplicaciones móviles y el servidor se hace de manera codificada y controlada.

Los resultados del desarrollo de cada aplicación móvil son los siguientes:

### **Aplicación Web Móvil de Referencias Médicas**

- Diseñada para un conjunto de usuarios del sistema SOS-TRIAJE (Médico Triage, Médico Especialista).
- Contempla un nuevo usuario (Médico Rural) con características y funcionalidades propias.
- Permite la comunicación de todos los usuarios con el Sistema SOS-TRIAJE de manera efectiva.
- Permite a los usuarios visualizar la información presentada en el Sistema SOS-TRIAJE dentro de sus dispositivos móviles.
- Ofrece todas las funcionalidades propias de cada usuario enmarcadas en un ámbito móvil, además de funcionalidades adicionales.
- Aporta beneficios de la comunicación móvil, independientemente del lugar en donde se encuentren los usuarios, pero limitativa en el uso de Internet.
- Logra beneficios de comunicación al tramitar de manera más efectiva las solicitudes.
- Complementa aún más, el Programa SOS Telemedicina para Venezuela, ahora con el desarrollo de tecnología móvil.

### **Aplicación Nativa n Android de Referencias Médicas**

- Permite la comunicación del usuario Médico Rural con el Sistema SOS-TRIAJE.
- Permite que el usuario realice todas sus funcionalidades y agrega otras dentro del sistema (crear caso, ver casos, ver respuestas, ver casos pendientes, sincronizar)
- Ofrece la ventaja de manejarse bajo un ambiente *stand alone* sin dependencia de Internet.
- Aporta los beneficios del desarrollo de un sistema operativo móvil de código abierto.
- Logra sincronizarse con la base de datos del SOS-TRIAJE y mostrar al usuario información consistente.
- Logra beneficios de comunicación al tramitar de manera más efectiva las solicitudes.
- Complementa aún más, el Programa SOS Telemedicina para Venezuela, ahora con el desarrollo de tecnología móvil.

## CAPÍTULO VI – CONCLUSIONES

El desarrollo de este trabajo dio como resultado el Sistema Móvil de Referencias Médicas, el cual se adapta al formato del Sistema SOS-TRIAJE, y adicionalmente agrega mejoras cumpliendo con su objetivo principal que es lograr la comunicación entre médicos rurales y especialistas para el desarrollo de diagnóstico a las patologías presentadas por los pacientes a través de los diferentes dispositivos móviles.

El desarrollo de éste sistema móvil está implementado en dos aplicaciones: una aplicación Web móvil y una aplicación nativa del Sistema Operativo Android, que permite a los médicos especialistas obtener información acerca del estado de salud de un paciente registrado en el sistema, así como el registro de nuevos casos cuya información no ha sido almacenada. Con esto se superan varias de las limitaciones de una comunicación tradicional a través de computadoras personales, y se lleva al mundo móvil permitiendo el proceso de comunicación desde el lugar en donde se encuentran y obteniendo los beneficios que ofrece la tecnología móvil.

Al desarrollar el Sistema Móvil de Referencias Médicas se logra que los médicos rurales puedan obtener de manera más rápida y eficiente segundas opiniones médicas sobre patologías presentadas por los pacientes desde los centros asistenciales.

El Sistema Móvil de Referencias Médicas colabora en uno de los enfoques fundamentales de la Telemedicina, específicamente del programa SOS Telemedicina como lo es contribuir con el desarrollo de una Red Nacional de Telemedicina, que proporcione e intercambie información a través de un sistema interconectado entre las distintas estructuras sanitarias a nivel nacional, ya que proporciona una interfaz móvil que permite lograr esta interconexión entre los centros asistenciales y los médicos especialistas sin la necesidad de estar situado frente a una computadora, y más aún, con la posibilidad de utilizar su dispositivo móvil personal como herramienta de trabajo.

Adicionalmente, con el desarrollo de este trabajo se le está dando la posibilidad a los profesionales de la salud que se encuentra en las comunidades rurales y sub-urbanas de difícil acceso, tener un servicio especializado de tele-diagnóstico desde dos frentes móviles: la aplicación Web móvil para los casos en que el médico rural tenga la posibilidad de contar con Internet inalámbrico (wi-fi) o con red de datos, o desde la aplicación nativa en Android cuando el médico no tiene la posibilidad de tener conexión a Internet, permitiendo que mejore la calidad

de atención a los pacientes y haciendo más fácil poder almacenar datos clínicos en el dispositivo móvil de forma segura e inmediata para luego ser transferidos a la base de datos al momento de tener conexión a datos.

En esta investigación se utilizó una adaptación de la metodología de desarrollo ágil «Programación Extrema», mejor conocido como «XP. La metodología de desarrollo ágil permitió la adaptabilidad del proyecto frente a los esperados cambios de requisitos sobre la marcha, además agilizó el desarrollo permitiendo redefinir la orientación del mismo luego de culminada cada iteración con la finalidad de satisfacer los requisitos restantes.

El patrón de diseño utilizado para este proyecto fue «Modelo Vista Controlador», conocido por sus sigas como «MVC». Como resultado se obtuvo un Sistema de Referencias Médicas para dispositivos móviles que es modular, con clara separación de la interfaz, los datos y la lógica del negocio lo que le permite ser mantenible, reutilizable, y de fácil modificación

En cuanto al diseño de las interfaces y del flujo de navegación del Sistema Móvil de Referencias Médicas es importante mencionar que fue necesario planificar los distintos entregables basándose en el Sistema SOS-TRIAJE. Una vez desarrolladas las aplicaciones se realizaron cambios para que se adapten al diseño definido lo que generó un esfuerzo elevado al momento de aplicar los cambios ya que cuando mientras más adelantados se está en las etapas de desarrollo del software, los cambios de alcance y/o funcionalidades pueden ser altamente costosos en cuanto a tiempo e impactan en casi todas las etapas del desarrollo del software. Se recomienda para futuros desarrollos definir con antelación el diseño y las funcionalidades requeridas antes de comenzar con el desarrollo de las funcionalidades de los sistemas.

Este sistema móvil de permite incluir mejoras o nuevas funcionalidades, entre las que se mencionan:

- Un módulo de Médico Rural para referencias médicas, que permite crear casos desde los centros asistenciales para enviarlos a la unidad de triaje.
- Posibilidad de utilizar la cámara del teléfono para capturar una foto del paciente, grabar un video o un audio, cargar un archivo de texto y permitiendo adjuntarlo a la creación del caso, con el fin de que la información levantada sea más completa al momento de registrar un caso. Se recomienda implementar la posibilidad de tratar archivos de audio, texto, video o algún otro formato diferente al de imágenes en el servidor y en el Sistema SOS-TRIAJE.

- Posibilidad de registrar y consultar casos sin conexión a Internet y luego sincronizarlos a la base de datos al momento de tener conectividad de datos o wi-fi.
- Posibilidad de que los usuarios Médico Especialista, Médico Rural y Médico Triage, puedan realizar cualquiera de sus funcionalidades de forma remota desde un dispositivo móvil aun y cuando no se encuentre en el centro asistencial, no limitando la atención a un horario específico.

El sistema en su versión web móvil es altamente portable y multiplataforma pudiendo ser utilizado desde cualquier dispositivo móvil que tenga un navegador web y conexión a internet elevando las posibilidades de acceso desde cualquier lugar en cualquier momento.

El desarrollo de un sistema móvil bajo el sistema operativo Android permitió aportar beneficios al Sistema de Referencias Médicas, ya que agrega funcionalidades al usuario y permite que el mismo no dependa de Internet. Android permite que la aplicación pueda ser instalada en casi cualquier dispositivo móvil, su mayor ventaja es que por ser de código abierto permitió a los desarrolladores obtener una amplia gama de soluciones para alcanzar los requerimientos planteados.

El sistema en su versión Android está desarrollado para que esté disponible desde la versión 2.2 de Android y superiores, es decir está disponible para todas las versiones del sistema operativo, ya que el 99% de los usuarios Android tienen versiones superiores a la versión 2.2 y de hecho la tendencia es la migración a las versiones 4.1 y superiores. Esto permite que la aplicación sea altamente portable y pueda ser utilizada en cualquier dispositivo con este sistema operativo.

Se tomó la decisión de desarrollar la aplicación nativa en Android a diferencia de los demás sistemas operativos móviles principalmente por su bajo costo de desarrollo, ya que el entorno de desarrollo es 100% gratuito y de software libre. Adicionalmente basado en nuestras investigaciones Android es el dispositivo más utilizado por encima de iOS, Blackberry y Windows Phone y Symbian, por lo que representa otra ventaja ya que la aplicación estará disponible para mayor cantidad de dispositivos. Otro factor que influyó en nuestra decisión es que al ser Android de código abierto, se encuentra disponible gran cantidad de documentación, librerías y códigos ejemplos en la web que pueden ser utilizados para el desarrollo de cualquier aplicación facilitando el aprendizaje y el desarrollo para este sistema operativo.

Es necesario destacar que fue invertida inicialmente gran cantidad de tiempo para analizar y comprender el funcionamiento de la herramienta Android SDK, así como la estructura y funcionamiento de una aplicación en Android ya que ninguno de los miembros del equipo de

desarrollo estaba familiarizado con el entorno. Esa inversión inicial repercutió en beneficios al momento de realizar modificaciones sobre la aplicación.

La solución software elaborada es el Sistema Móvil de Referencias Médicas, está compuesta por las aplicaciones: Aplicación Web Móvil de Referencias Médica y Aplicación Nativa en Android de Referencias Médicas son un primer paso para acercar un servicio especializado de salud de calidad, como lo es permitir a los médicos rurales obtener de manera más rápida y eficiente segundas opiniones médicas sobre patologías presentadas por los pacientes de poblaciones rurales del país donde hace presencia el programa SOS Telemedicina para Venezuela.

Se recomienda como futuras mejoras:

- Implementar el manejo de archivos de voz en el Sistema de Referencias Médicas, con el fin de enriquecer los sistemas con el uso de contenido multimedia.
- Integrar en la aplicación nativa en Android los demás usuarios del sistema con sus respectivas funcionalidades, y ampliar así el uso de esta aplicación.
- Desarrollar la misma aplicación para otros sistemas operativos usados como IOS, Windows Phone y Blackberry y abarcar así más tecnologías existentes.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] **Proyecto SOS Telemedicina para Venezuela.** [En línea]  
<http://www.sos.ucv.ve/SOSTelemedicina.pdf>. 2007. Tesis de grado. UCV, Facultad de Ciencias.
- [2] **Gómez, Angélica. “Apoyo Tecnológico en el diseño, desarrollo, instalación y configuración de aplicaciones, Programa SOS Telemedicina para Venezuela”.** Tesis de grado. Universidad Católica Andrés Bello. Febrero de 2013.
- [3] **Dispositivos Móviles.** [En Línea] 2013.  
<https://sites.google.com/site/mliburutegia10/dispositivos-moviles>
- [4] **Telefonía Móvil.** [En Línea] 2012  
[.http://es.wikipedia.org/wiki/Telefon%C3%ADa\\_m%C3%B3vil](http://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil)
- [5] **Smartphone.** [En Línea] Noviembre de 2012  
[http://smartthechnology.blogspot.com/2012\\_11\\_01\\_archive.html](http://smartthechnology.blogspot.com/2012_11_01_archive.html)
- [6] **Calero Monteagudo, Antonio. “Tecnologías Móviles con Java”.** [En Línea] Noviembre de 2011. <http://www.iti.es/media/about/docs/tic/08/articulo1.pdf>
- [7] **Sistemas Operativos Móviles.** [En Línea] Octubre de 2012.  
<http://generacioninteligente.blogspot.com/2012/10/sistemas-operativos-moviles.html>
- [8] **Sistemas Operativos para celulares.** [En Línea] 2011  
[http://www.slideshare.net/diego\\_morales/sistemas-operativos-para-celulares-11136511](http://www.slideshare.net/diego_morales/sistemas-operativos-para-celulares-11136511)
- [9] **Conocer Nuevas Tecnologías.** [En Línea] 2012  
<http://www.infouruguay.com.uy/CONOCER-NUEVAS-TECNOLOGIAS.htm>
- [10] **La Tecnología. La Era Tecnológica.** [En Línea] Noviembre de 2012  
<http://issuu.com/cindyarciniegas/docs/latecnologia>
- [9] **Sistemas Operativos Móviles.** [En Línea] Enero de 2012  
<http://www.colombiadigital.net/entorno-tic/especial-del-mes/dispositivos-moviles/item/1341-sistemas-operativos-m%C3%B3viles.html>

[10] Prieto Padrón, Armando y Rodríguez León, Ángel. “Sistema De Historia Médica Electrónica Para SOS Telemedicina”. Mayo del 2012. Tesis de grado. UCV, Escuela de Computación.

[11] Colman, Marisa Roxana y Negri, Gabriel Alejandro. “Una aplicación móvil de realidad aumentada en el ámbito universitario”. [En línea] Noviembre de 2011.  
<http://revista.info.unlp.edu.ar/tesinas/tesis112.pdf>

[12] Native App vs. Mobile Web App: A Quick Comparison. [En línea] Julio 2012  
<http://sixrevisions.com/mobile/native-app-vs-mobile-web-app-comparison/>

[13] Metodologías de Desarrollo de Software. [En línea] Febrero 2012  
<http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>

[14] Procesos de Desarrollo: RUP, XP y FDD. [En línea] Febrero 2002  
<http://www.willydev.net/descargas/Articulos/General/cualxpfd Drupal.PDF>

[15] Metodología XP Vs. Metodología RUP. [En línea] 2014.  
<http://metodologiaxpvsmetodologiarup.blogspot.com/>

[16] Unified Modeling Language (UML). [En línea] 2013. <http://www.uml.org/>

[17] Modelo Vista Controlador. [En línea] 2014.  
<http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas>

[18] HTML5. [En línea] 2014. <http://www.w3.org/TR/html5/>

[19] Introducción a JavaScript. [En línea] 2012. <http://librosweb.es/javascript/>

[20] jQuery. Category: Ajax. [En línea] 2014. <https://api.jquery.com/category/ajax/>

[21] jQuery. [En línea] 2014. <http://jquery.com/>

[22] jQuery Mobile . [En línea] 2014. <http://jquerymobile.com/>

[23] PHP . [En línea] 2014. <http://php.net>

- [24] **Casillas, Alberto y Pérez, Oscar.** “Bases de datos”. [En Línea] 2012.  
[http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06\\_M2109\\_02151.pdf](http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02151.pdf)
- [25] **PHP: WampServer definición, instalación y configuración.** [En Línea] 2013  
<http://codegeando.blogspot.com/2013/03/php-wampserver-definicion-instalacion-y.html>
- [26] **Molina, Yeicy; Sandoval, Jonathan y Toledo, Santiago.** “Sistema operativo Android: Características y Funcionalidad para Dispositivos Móviles”. Marzo 2012. [En Línea]  
<http://repositorio.utp.edu.co/dspace/bitstream/11059/2687/1/0053M722.pdf>
- [27] **Martínez, Jorge.** “Fundamentos de Programación en Java”. Junio 2013. [En Línea]  
<http://pendientedemigracion.ucm.es/info/tecnomovil/documentos/fjava.pdf>
- [28] **Android Development Tutorial.** Febrero 2012. [En Línea]  
[http://www.cse.nd.edu/~cpoellab/teaching/cse40816/android\\_tutorial.pdf](http://www.cse.nd.edu/~cpoellab/teaching/cse40816/android_tutorial.pdf)
- [29] **Introducción a XML.** Mayp 2010. [En Línea] <http://quark.fe.up.pt/cursoxml/curso.pdf>
- [30] **Introducción a Json.** Agosto 2011. [En Línea] <http://www.json.org/json-es.html>
- [31] **SQLite Tutorial.** Agosto 2011. [En Línea] [http://cs.indstate.edu/CS473/sqlite\\_tutorial.pdf](http://cs.indstate.edu/CS473/sqlite_tutorial.pdf)
- [32] **Labra, José.** “Introducción a los Servicios Web”. Octubre 2009. [En Línea]  
<http://di002.edv.uniovi.es/~labra/cursos/XML/ServiciosWeb.pdf>
- [33] **Threads, AsyncTasks & Handlers.** [En Línea]  
[https://www.cs.umd.edu/class/fall2011/cmsc436/CMSC436/Lectures\\_Labs\\_files/ThreadsAndMessaging.pdf](https://www.cs.umd.edu/class/fall2011/cmsc436/CMSC436/Lectures_Labs_files/ThreadsAndMessaging.pdf)
- [34] **Gartner Group.** [En Línea] <http://www.gartner.com/technology/home.jsp>

## APÉNDICE A – DOCUMENTACIÓN GRÁFICA

A continuación se expone de manera detallada la documentación gráfica del sistema móvil de Referencias Médicas.

### Diagramas de casos de uso de la aplicación Web móvil de Referencias Médicas

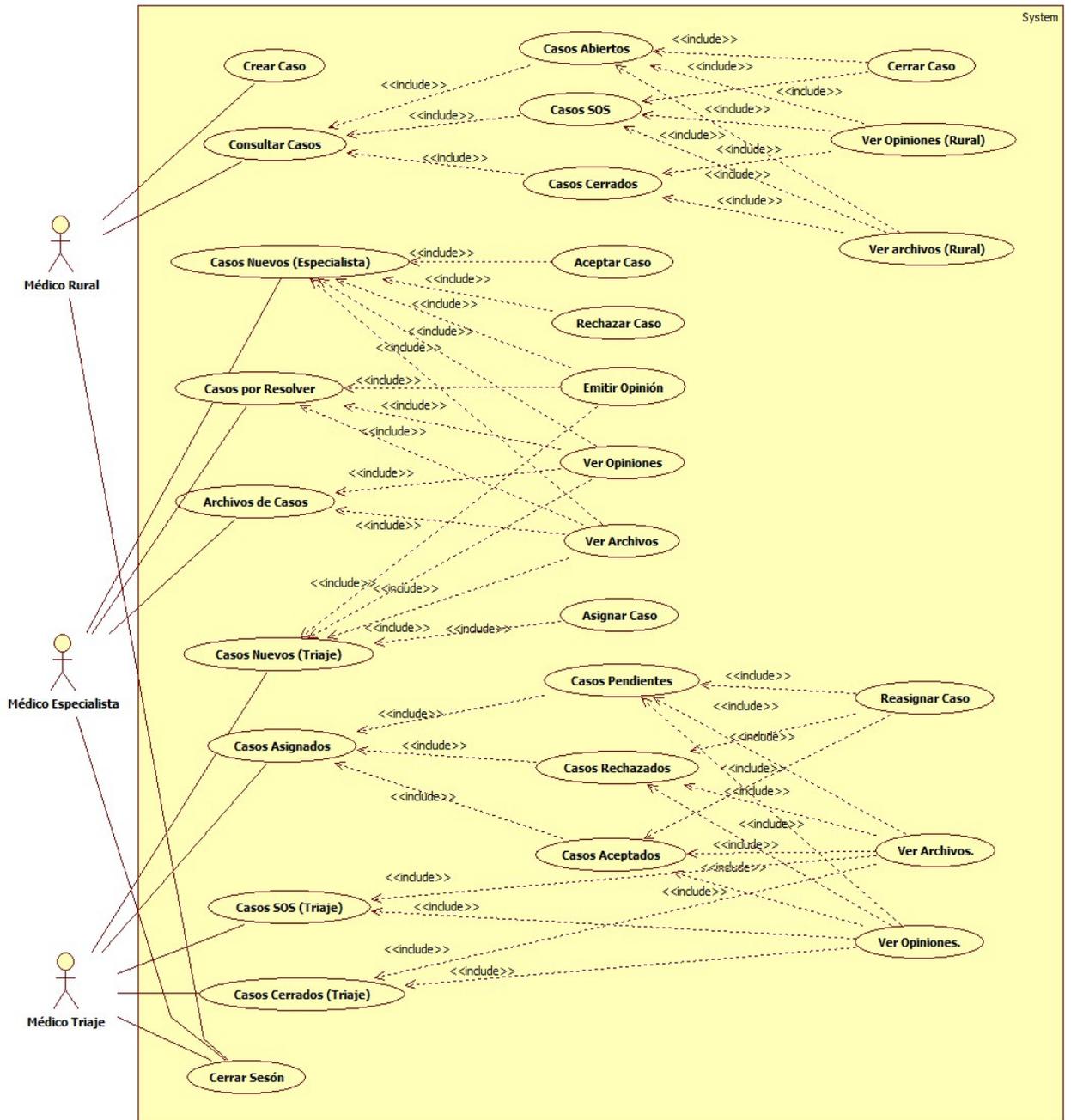


Figura A.1. Diagrama de caso de uso Nivel 2

En la Figura A.1 se puede apreciar el diagrama de Casos de Uso Nivel 2, con todos los casos asociados a los diferentes actores del sistema y sus respectivas herencias.

### Diagramas de actividad del sistema móvil de Referencias Médicas

- **Crear Caso**

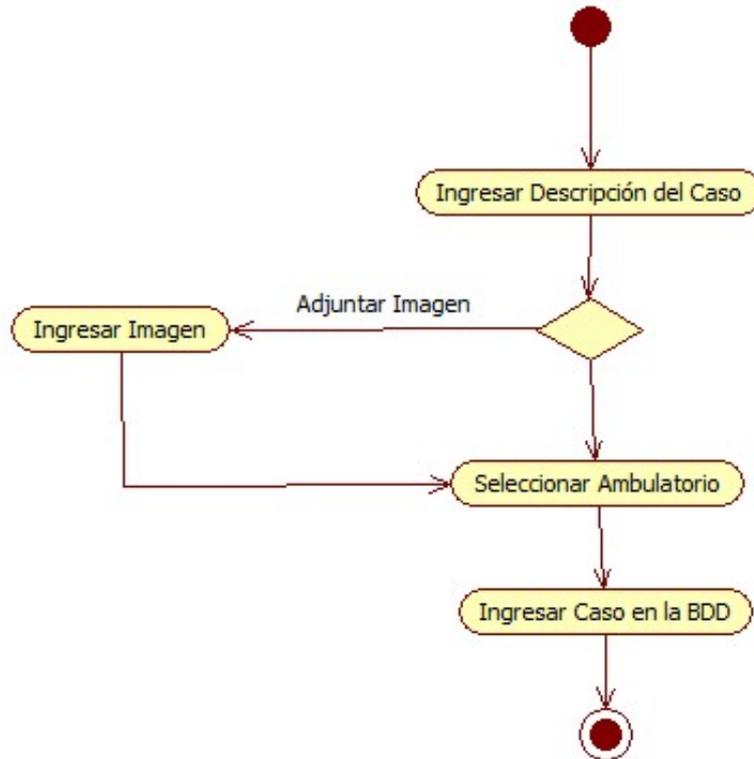


Figura A.2: Diagrama de actividad para el caso de uso "Crear Caso"

En la Figura A.2 se puede apreciar el diagrama de actividad para el caso de uso "Crear Caso", tomando en cuenta el flujo de evento típico y los flujos de eventos excepcionales de esta actividad dentro del sistema.

- Consultar Casos

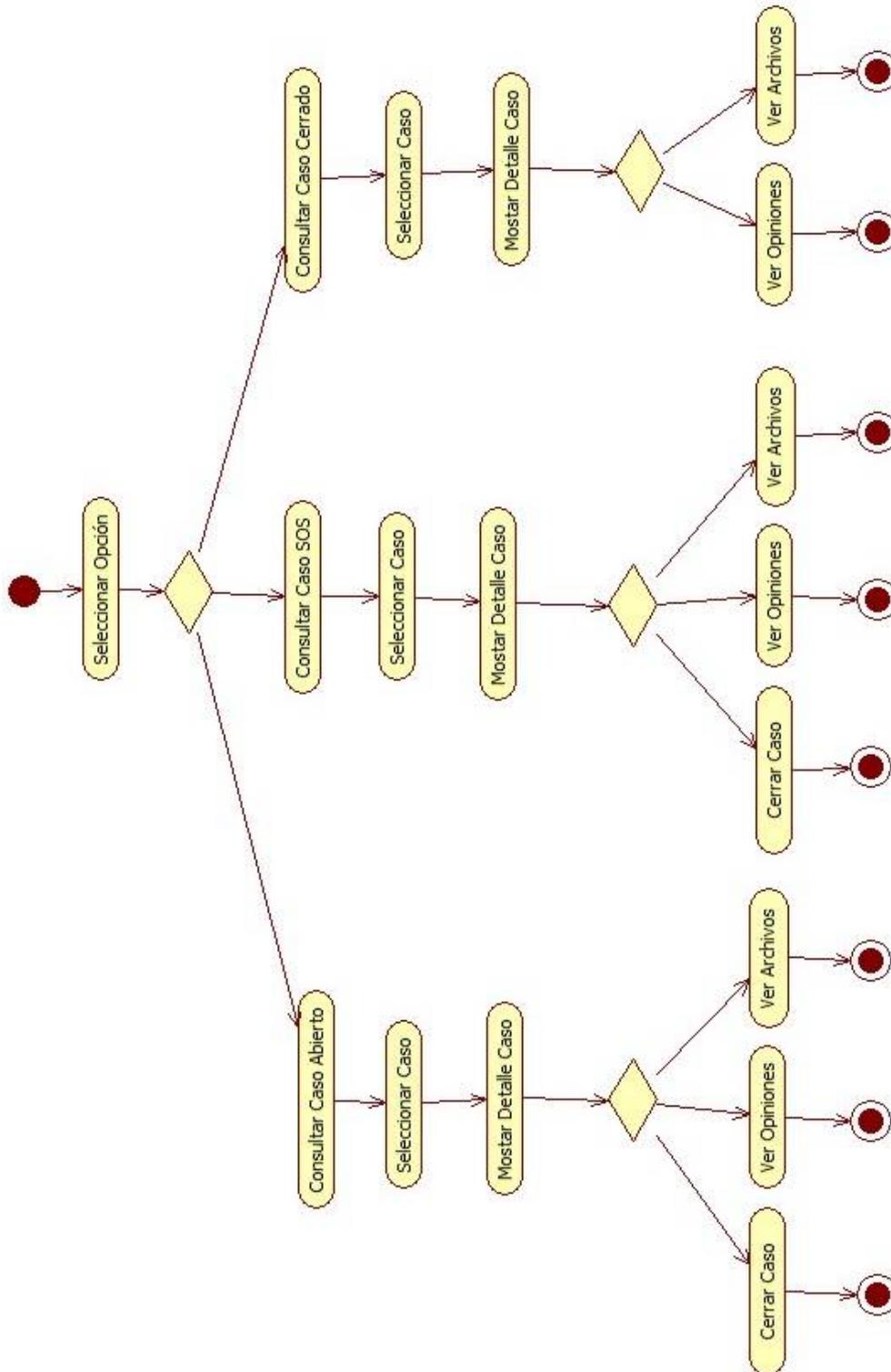


Figura A.3. Diagrama de actividad para el caso de uso “Consultar Casos”

En la Figura A.3 se puede apreciar el diagrama de actividad para el caso de uso “Consultar Caso”, tomando en cuenta los diferentes flujos de eventos típicos y excepcionales acerca de todas las acciones para esta actividad dentro del sistema.

- **Casos por Resolver**

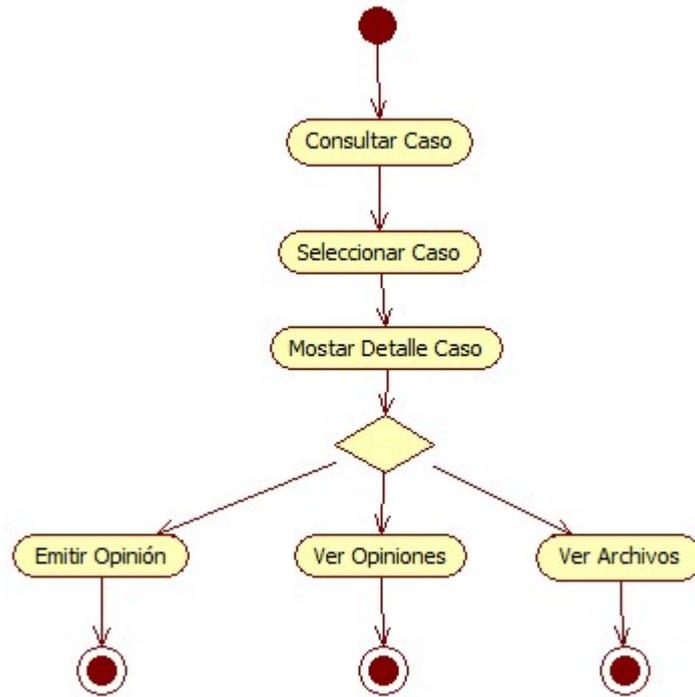


Figura A.4. Diagrama de actividad para el caso de uso “Casos por Resolver”

En la Figura A.4, se puede apreciar el diagrama de actividad para el caso de uso “Casos por Resolver”, tomando en cuenta el flujo de evento típico para esta actividad dentro del sistema.

- **Consultar Casos Nuevos**

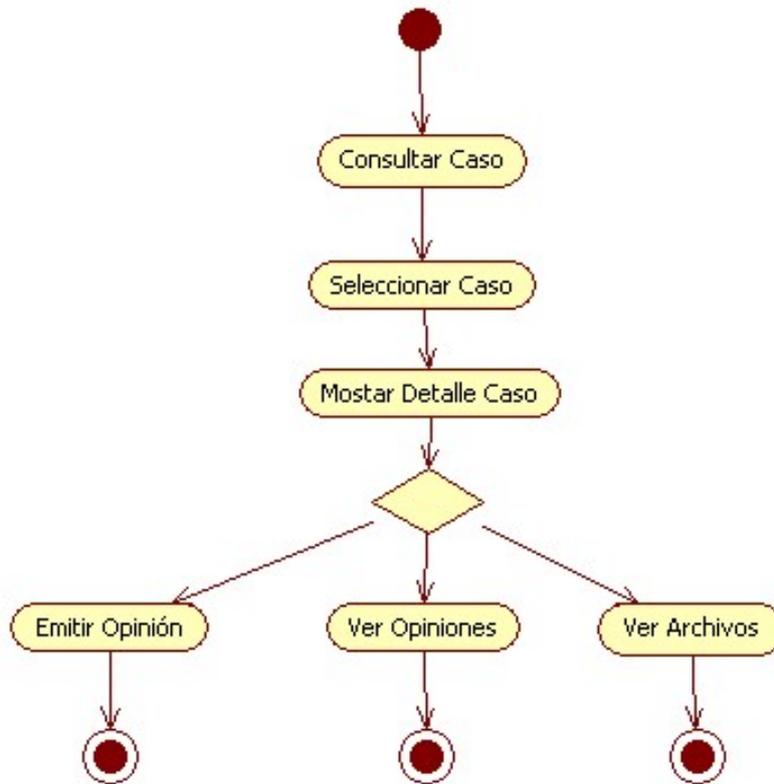


Figura A.5. Diagrama de actividad para el caso de uso “Consultar Casos Nuevos”

En la Figura A.5 se puede apreciar el diagrama de actividad para el caso de uso “Consultar Casos Nuevos”, tomando en cuenta los diferentes flujos de eventos que intervienen esta actividad.

- **Consultar Casos SOS y Casos Cerrados**

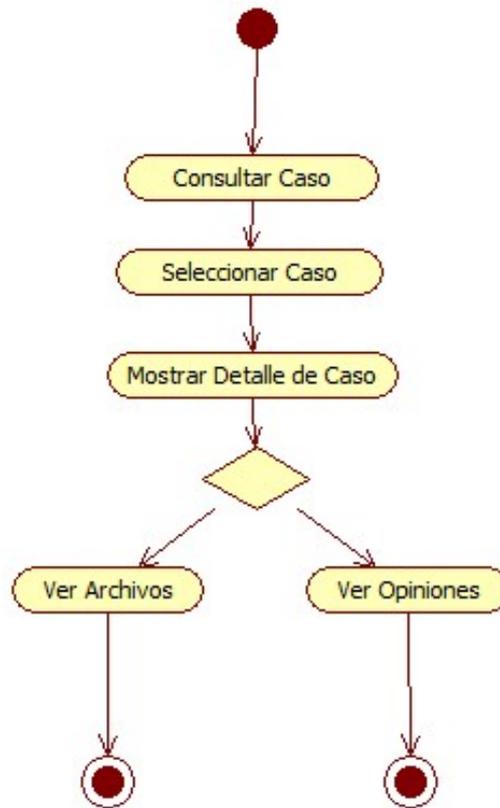


Figura A.6. Diagrama de actividad para los casos de uso “Consultar Casos SOS” y “Consultar Casos Cerrados”

En la Figura A.2 se puede apreciar el diagrama de actividad para los casos de uso “Consultar Casos SOS” y “Consultar Casos Cerrados”, tomando en cuenta los diferentes flujos de eventos y acciones que pueden conllevar esta actividad dentro del sistema.

- **Asignar/Reasignar Casos**



Figura A.7: Diagrama de actividad para los casos de uso “Asignar Caso” y “Reasignar Caso”

En la Figura A.7, se puede apreciar el diagrama de actividad para los casos de uso “Asignar Caso” y “Reasignar Caso” tomando en cuenta el flujo de evento típico que realiza esta actividad dentro del sistema.

- **Emitir Opinión**

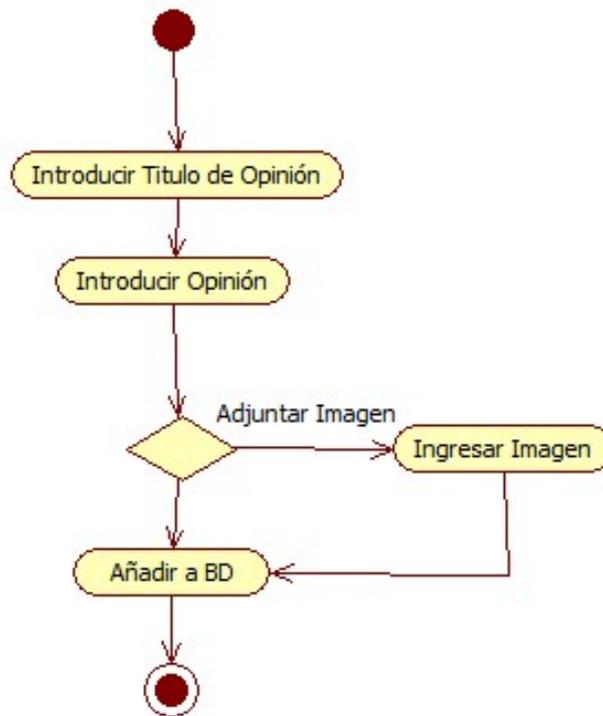


Figura A.8: Diagrama de actividad para el caso de uso "Emitir Opinión"

En la Figura A.8, se puede apreciar el diagrama de actividad para el caso de uso "Emitir Opinión", tomando en cuenta el flujo de evento típico y los flujos de eventos excepcionales de esta actividad dentro del sistema.

- **Salir del Sistema**

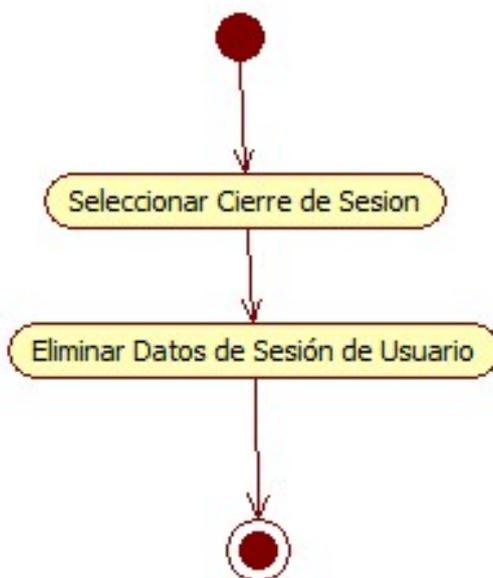


Figura A.9. Diagrama de actividad para el caso de uso "Salir del Sistema"

En la Figura A.9, se puede apreciar el diagrama de actividad para el caso de uso "Salir del Sistema", tomando en cuenta el flujo de evento típico de esta actividad de fin de sesión dentro del sistema.

## Máquina de estados del sistema móvil de Referencias Médicas

A continuación en la Tabla A.1, se detallan los componentes la máquina de estados del Sistema Móvil de Referencias Médicas.

**Tabla A.1 Componentes de la máquina de estados y su descripción**

| <b>ESTADO</b>               | <b>DESCRIPCIÓN</b>  |
|-----------------------------|---|
| <b>En espera</b>            | Indica que el caso se encuentra en espera de asignación y/o resolución.     |
| <b>Asignado</b>             | Indica que el caso ha sido asignado a un especialista.                      |
| <b>En proceso 1er nivel</b> | Indica que el caso ha sido asignado a un especialista.                      |
| <b>Segunda opinión</b>      | Indica que el especialista solicitó una segunda opinión.                    |
| <b>En proceso 2do nivel</b> | Indica que la solicitud de segunda opinión ha sido aceptada para responder. |
| <b>Resuelto 2do nivel</b>   | Indica que la segunda opinión fue resuelta.                                 |
| <b>Resuelto 1er nivel</b>   | Indica que el caso ha sido resuelto.  |
| <b>Cerrado</b>              | Indica que el caso ha sido cerrado y se enviará finalmente a HME.           |
| <b>Reasignado</b>           | Indica que el caso ha sido reasignado por límite de tiempo en espera.       |
| <b>Rechazado 1er nivel</b>  | Indica que ha sido rechazada la asignación del caso.                        |
| <b>Rechazado 2do nivel</b>  | Indica que ha sido rechazada la solicitud de segunda opinión.               |

