

UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS EMPRESARIALES
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y
TECNOLOGÍA DE LA INFORMACIÓN
CAMPUS CHILLÁN



**“DESARROLLO DE APLICACIÓN MÓVIL SOBRE
PLATAFORMA ANDROID EN APOYO A VISITAS
MÉDICAS”**

**VERÓNICA ANDREA RAMÍREZ NORAMBUENA
CHRISTOPHER EDGARDO AREVALO GONZALES**

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO DE EJECUCIÓN EN COMPUTACIÓN E INFORMÁTICA**

Chillán, Agosto 2010

UNIVERSIDAD DEL BIO-BIO
FACULTAD DE CIENCIAS EMPRESARIALES
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y
TECNOLOGÍA DE LA INFORMACIÓN
CAMPUS CHILLÁN

**“DESARROLLO DE APLICACIÓN MÓVIL SOBRE
PLATAFORMA ANDROID EN APOYO A VISITAS
MÉDICAS”**

**VERÓNICA ANDREA RAMÍREZ NORAMBUENA
CHRISTOPHER EDGARDO AREVALO GONZALES**

PROFESOR GUIA : SR. LUIS GAJARDO DÍAZ
PROFESOR INFORMANTE : SRTA. MARIA ANTONIETA SOTO CHICO
NOTA FINAL EXAMEN TITULO : _____

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO DE EJECUCIÓN EN COMPUTACIÓN E INFORMÁTICA**

Chillán, Agosto 2010

Agradecimientos

Agradezco a Dios y a mi abuelita que desde el cielo iluminaron mi camino y guiaron mis pasos. Agradecer a mi madre por el cariño, la paciencia, por alegrarse en cada uno de mis triunfos y contenerme en cada una de mis derrotas. A mi padre por el esfuerzo de financiar mi carrera. A mi "colega", hermano sin lugar a dudas fuiste mi modelo a seguir. A Juan Carlos por el amor, dedicación y apoyo incondicional, gracias por hacer soleado hasta el día más nublado.

Verónica Ramírez Norambuena

Quisiera dar las gracias a Dios, a mi Familia, Profesores, Amigos y Compañeros por el eterno e incondicional apoyo que me han brindado, sin el cual jamás habría logrado concretar esta etapa crucial de mi vida.

Christopher Arévalo Gonzáles

Agradecemos a nuestro profesor Luis Gajardo, por dedicar gran parte de su tiempo en guiar el desarrollo de este proyecto. A nuestros compañeros quienes hicieron grata la estadía en la Universidad.

Los autores

Dedicatoria

*Dedico este trabajo a mi madre María Angélica
por hacer de mi todo lo que soy.*

Verónica Ramírez Norambuena

*A mis padres, humilde ejemplo de sencillez,
que me enseñaron todo.*

Christopher Edgardo Arévalo González

RESUMEN

El presente informe detalla el desarrollo de una aplicación móvil para médicos que realizan visitas domiciliarias a sus pacientes, tanto para los independientes como los que trabajan bajo el alero del estado en consultorios públicos u hospitales.

La idea principal de esta aplicación móvil es permitir a los médicos tratantes administrar sus visitas, brindándoles información de ellas y de los pacientes relacionados. Además disponer de servicios asociados tales como ver en el mapa la dirección de un paciente, ver los historiales de observaciones, crear fichas médicas de pacientes no existentes, agendar y ver detalle de visitas médicas.

La novedad de esta aplicación es que su desarrollo fue realizado en la nueva plataforma para dispositivos móviles inteligentes (SmartPhone) Android de Google, la cual fue creada hace tres años atrás por AndroidInc y comprada posteriormente por Google. Esta plataforma permite la creación de aplicaciones sin incurrir en costos de licencias, puesto que es gratuita.

Además se implementó otra aplicación, con la estructura de un servicio la cual se encuentra albergada en un servidor, ya que está encargada de responder a través del uso de socket cualquier solicitud proveniente de la agenda médica, de esta forma los datos son accedidos y almacenados en el servidor.

El modelo de desarrollo utilizado es el basado en el modelo incremental, utilizando la metodología orientado a objetos con el uso del patrón de diseño Modelo Vista Controlador (MVC).

Este proyecto permitirá a los médicos administrar sus visitas médicas de una manera más ordenada, eficiente, oportuna e innovadora, accediendo a la información necesaria del paciente en cualquier lugar, reduciendo también considerablemente el tiempo de ingresos de fichas y traspaso de información de visitas.

Invitamos al lector a introducirse en el proyecto de la agenda médica y conocer además un poco más de la plataforma Android y su funcionamiento.

ÍNDICE GENERAL

Capítulo I

INTRODUCCIÓN	13
1.1. INTRODUCCIÓN GENERAL	14
1.2. SITUACIÓN ACTUAL	15
1.2.1. Registrar una nueva ficha médica	15
1.2.2. Anexar información a una ficha médica	17
1.2.3. Conocer las citas asignadas	18
1.2.4. Ubicar la dirección de los pacientes que serán visitados a domicilio	19
1.3. DESCRIPCIÓN DEL PROBLEMA.....	21
1.3.1. Situaciones que promueven la atención médica a domicilio.	21
1.4. OBJETIVOS GENERALES Y ESPECÍFICOS.....	23
1.4.1. Objetivo General	23
1.4.2. Objetivos específicos	24
1.5. METODOLOGÍA A UTILIZAR.....	24
1.6. TRABAJOS RELACIONADOS.....	25
1.6.1. Trabajos relacionados en la Universidad	25
1.6.2. Trabajos relacionados fuera del ámbito académico.....	26

Capítulo II

TECNOLOGÍA PARA DISPOSITIVOS MÓVILES	27
2.1. TELEFONÍA MÓVIL ACTUAL EN CHILE	28
2.2. ECOSISTEMA MÓVIL	30
2.2.1 Operadores de telefonía móvil	30
2.2.2. Redes.....	32
2.2.2.1 Redes de telefonía móvil (PLMN)	32
2.2.2.2 Sistema GSM (Global System for Mobile communications)	34
2.2.2.3. Tecnologías de telefonía móvil.....	36
2.2.3. Dispositivos Móviles	37
2.2.3.1. Características principales HTC Magic	39
2.2.4. Plataformas	40
2.2.4.1. Framework de aplicaciones	42
2.2.5. Servicios	44

Capítulo III

ANDROID	45
3.1. HISTORIA DE ANDROID.....	46
3.2. CARACTERÍSTICAS TÉCNICAS DE ANDROID.....	47
3.2.1. Arquitectura de Android.....	48
3.3. FILOSOFÍA DE DESARROLLO ANDROID	50
3.3.1. Modelo Vista Controlador	50
3.3.2. Lenguaje XML	51
3.3.3. Lenguaje Java	52

Capítulo IV

DESARROLLO DE APLICACIÓN NATIVA SOBRE ANDROID	53
4.1. ANÁLISIS DE REQUISITOS	54
4.1.1. Requisitos no funcionales.....	54
4.1.2. Requisitos funcionales	55
4.1.3. Plantilla combinada	57
4.1.4. Identificación de actores del sistema	59
4.1.5. Modelo de casos de Uso.....	60
4.1.5. Especificación de casos de uso	61
4.2 DISEÑO DE PROTOTIPOS DE USABILIDAD	71
4.3. MODELO ENTIDAD RELACIÓN.....	77
4.4. MODELO CONCEPTUAL	78
4.5. DIAGRAMA DE DESPLIEGUE	79
4.7. DIAGRAMAS DE CLASE DEL DISEÑO	80
4.7. DIAGRAMAS DE COLABORACIÓN	81
4.8. DIAGRAMA DE CLASES	85
4.8. PATRONES DE DISEÑO	88
4.8.1. Patrón Modelo vista controlador (MVC)	88
4.8.2. Singleton	89
4.8.3. Patrón DAO.....	89
4.9. MAPAS DE NAVEGACIÓN	90

Capítulo V

IMPLEMENTACIÓN_Y PRUEBAS.....	92
5.1. DESARROLLO DE LA INTERFAZ.....	93
5.2. USO DE MAPAS PARA LA LOCALIZACIÓN	95

5.2.1. Obtención de la API Key	95
5.2.2. Implementación de la Actividad que muestra el mapa.	96
5.2.3. Dibujar una dirección en el mapa.....	99
5.3. PRUEBAS DE USABILIDAD.	100
5.3.1. Acciones a medir	101
5.3.2. Resultados de las pruebas de usabilidad	102
5.4. PRUEBAS DE UNIDAD, DESEMPEÑO Y TENSIÓN.....	104
5.4.1. Pruebas de unidad	104
5.4.2. Pruebas de desempeño y tensión.....	107
5.5. DETALLES DE LA PERSISTENCIA DE DATOS	109
5.5.2. Detalle de tablas	110
5.6. SEGURIDAD DE LA APLICACIÓN	114
5.6.1. Código QR.	114
5.6.2. Implementación del inicio de sesión.	115
5.6.3. Generación del código QR.	116
 Capítulo VI	
INTERACCIÓN CON EL SERVIDOR WEB	117
5.1. SOCKET PARA LA INTERACCIÓN	118
5.2. IMPLEMENTACIÓN DE LOS SOCKETS.....	118
 CONCLUSIONES	122
TRABAJOS FUTUROS	124
BIBLIOGRAFÍA	125
ANEXOS	126
ANEXO A: Centro de atención primaria federico puga borne.....	126
ANEXO B: Estructura ficha médica.....	128
Anexo C: Ejemplo mini aplicación en Android.....	129
Anexo D: Encuesta proceso de atención médica de pacientes a domicilio.....	131
Anexo E: Prueba de usabilidad	132

ÍNDICE DE FIGURAS

Figura 1, Diagrama de Actividad de registrar nueva ficha médica.....	16
Figura 2, Diagrama de actividad anexar información a una ficha médica.....	17
Figura 3, Diagrama de actividad ver citas asignadas.	18
Figura 4, Diagrama de actividad ubicar dirección de pacientes.....	20
Figura 5, Gráfico de crecimiento anual en telefonía móvil.....	29
Figura 6, Gráfico de preferencias de servicio	29
Figura 7, Capas ecosistema móvil.	30
Figura 8, Posición en el mercado de operadores móvil	31
Figura 9, Esquema de Red de telefonía móvil	32
Figura 10, Disposición de celdas celulares	35
Figura 11, HTC Magic	38
Figura 12, Arquitectura de Android	48
Figura 13, Diagrama de casos de uso.	60
Figura 14, Interfaz de ingreso a la aplicación	72
Figura 15, Componentes de usabilidad	73
Figura 16, Presentación de datos	74
Figura 17, Interfaz de ficha médica	75
Figura 18, Representación de dirección en el Mapa	76
Figura 19, Modelo Entidad Relación.....	77
Figura 20, Modelo Conceptual	78
Figura 21, Diagrama de despliegue.....	79
Figura 22, Diagrama de clase del diseño.....	80
Figura 23, Diagrama Escuchar peticiones.....	81
Figura 24, Diagrama Atender Cliente.....	82
Figura 25, Diagrama Conectar.....	82
Figura 26, Diagrama estaConectado.....	83
Figura 27, Diagrama esValido.....	83
Figura 28, Diagrama obtenerMedico.....	84
Figura 29, Diagrama cerrarConexion	84
Figura 30, Diagrama de Clases Package persistencia	85
Figura 31, Diagrama de clases package comunicación	86
Figura 32, Diagrama de clases agenda medica	87

Figura 33, Mapa de navegación	90
Figura 34, Código Ejemplo1	93
Figura 35, Vista Ejemplo 1	94
Figura 36, Obtención API Key	95
Figura 37, Uso API Key	96
Figura 38, Objetos básicos del mapa	97
Figura 39, Tipos de vista de Mapa	97
Figura 40, Dirección a mostrar en el mapa	98
Figura 41, Permiso para uso de la biblioteca GoogleMaps	98
Figura 42, Permiso para el uso de Internet	98
Figura 43, clase HelloItemizedOverlay	99
Figura 44, Implementación Overlay	99
Figura 45, Aplicación prueba de usabilidad	102
Figura 46, Módulo despliegue y búsqueda	107
Figura 47, Módulo autenticar medico	108
Figura 48, Módulo ingresar visita	108
Figura 50, Llamado a Barcode Scanner	115
Figura 51, Resultado QR	115
Figura 52, Página generadora de códigos QR	116
Figura 53, Inicializa socket	119
Figura 54, Crear Socket	119
Figura 55, Recibe peticiones	120
Figura 56, Atender petición	120
Figura 57, Petición al servidor	121

Índice de Tablas

Tabla 1, Requisitos no funcionales	54
Tabla 2, Requisitos Funcionales	56
Tabla 3, Plantilla combinada.....	58
Tabla 4, Actores del sistema	59
Tabla 5, Autenticar usuario	61
Tabla 6, Ver visitas del día	62
Tabla 7, Ver ficha médica	62
Tabla 8, Buscar citas por fecha.....	63
Tabla 9, Ver historial de observaciones.....	63
Tabla 10, Crear ficha médica.....	64
Tabla 11, Agregar observación medica	65
Tabla 12, Cerrar sesión.	65
Tabla 13, Buscar Dirección.....	66
Tabla 14, Llamar un paciente	66
Tabla 15, Marcar estados de visitas.....	67
Tabla 16, Buscar paciente.....	68
Tabla 17, Ver perfil del paciente	69
Tabla 18, Agregar visita médica.....	70
Tabla 19, Resultado prueba de usabilidad	103
Tabla 20, Prueba Autenticar Usuario	104
Tabla 21, Crear ficha médica.....	105
Tabla 22, Prueba ver dirección en el mapa.....	105
Tabla 23, Prueba buscar visitas por fecha	106
Tabla 24, Agregar visita médica.....	106
Tabla 25, Tabla Paciente.....	110
Tabla 26, Tabla Médico.....	110
Tabla 27, Tabla Cita	111
Tabla 28, Tabla observacion.....	112
Tabla 29, Tabla fichamedica.....	113

Capítulo I

INTRODUCCIÓN

Para comprender de qué trata un proyecto es necesario explicar detalladamente algunos aspectos importantes que rodean al problema que se desea solucionar.

Este primer capítulo aborda temas que interiorizan al lector con los objetivos del proyecto, destacando descripciones de la situación a la que están expuestos los médicos en terreno hoy en día y la problemática que le surge al desarrollar ciertas actividades en sus visitas domiciliarias.

Adicionalmente se incluyen los objetivos generales y específicos que tiene el proyecto, algunos trabajos realizados con anterioridad por otras personas o instituciones y la metodología seleccionada para el desarrollo.

1.1. INTRODUCCIÓN GENERAL

Hoy en día, el uso de las tecnologías ha cobrado vital importancia en la vida de las personas, ya que fueron creadas para simplificar un sinnúmero de actividades que a diario realizan las personas, las organizaciones y empresas. De esta forma es bastante común que una vez adoptadas en nuestro diario vivir sea difícil deshacerse o prescindir de ellas.

Como la tecnología siempre está en constante movimiento y su evolución viene ya siendo parte de su naturaleza no es extraño encontrarse en el mercado con productos innovadores y al alcance del bolsillo de personas de distintos estratos sociales, un claro ejemplo de ello es analizar el hecho de que en Chile existe aproximadamente un teléfono por persona.

También es cierto que las personas de la actualidad son más exigentes, pretenden controlar el máximo de actividades de la forma más cómoda que se pueda y con la menor cantidad de herramientas posibles. Es por esta última razón que se ve la salida al mercado de dispositivos que cumplen más de una función al mismo tiempo. Un ejemplo de ello son los computadores portátiles los cuales centralizan variadas actividades con la finalidad de mejorar la calidad de vida de las personas. Siguiendo esta línea se encuentran los teléfonos celulares que ya no ofrecen solo el servicio de realizar llamadas o enviar mensajes de texto a otros dispositivos, sino que cuentan con muchas más herramientas que hacen más atractivo el producto entre los cuales esta el GPS, cámara fotográfica, reproductor de música y vídeo, WIFI, Bluetooth entre otros. Permiten también ofrecer servicios parecidos a los que puede entregar un computador pero todo esto bajo algunas restricciones como lo pueden ser la capacidad, velocidad, etc., pero a pesar de eso se puede contar con aplicaciones que van en ayuda de ciertas actividades que un usuario en específico requiera.

Es bajo el concepto expuesto anteriormente que se desea apoyar a médicos que realizan visitas a domicilio mediante una aplicación instalada en un dispositivo móvil. Lo anterior permitirá agilizar la gestión de reservas de atención médica a pacientes en instituciones del área de la salud, tales como clínicas, hospitales u consultorios que proveen tratamientos a domicilio.

1.2. SITUACIÓN ACTUAL

La atención de pacientes a domicilio es uno de los servicios que demanda mayor esfuerzo en una institución médica, pues requiere coordinar múltiples recursos como el personal, utensilios y transporte. Su complejidad se puede apreciar observando instituciones del sector público donde la cantidad de pacientes va en aumento y el presupuesto es limitado.

Lo que se espera desarrollar es una solución que sea fácil de implementar en cualquier institución del área de la salud para agilizar el servicio, independiente de las normas y procedimientos que le gobiernen. La condición anterior implica realizar un análisis de los aspectos necesarios para llevar a cabo el procedimiento, por lo que se ha recurrido al consultorio Dr. Federico Puga Borne (ver Anexo A) para adquirir un conocimiento más detallado.

1.2.1. Registrar una nueva ficha médica

El registro de una ficha médica (ver Anexo B) es un proceso en el que intervienen principalmente tres entidades:

- 1) El paciente interesado en registrar su ficha médica.
- 2) Un administrativo del Servicio de Orientación Médica y Estadística (SOME).
- 3) Sistema de información que organiza las fichas médicas ya registradas.

La intervención de este último se debe a que las fichas actualmente están soportadas en papel, agrupadas en carpetas y guardadas en estantes, se calcula un total de 1.647 fichas haciendo poco viable cualquier procedimiento manual.

El procedimiento inicia con la solicitud de un paciente hacia un administrativo del sector SOME ubicado en las dependencias del consultorio. Primero se verifica que la ficha no haya sido registrada anteriormente, lo que se logra mediante una búsqueda en el sistema que organiza las fichas ya registradas, tras ingresar el Run del paciente, el sistema indicará si este fue o no ingresado. En el caso de no estarlo, el administrativo procederá a solicitar los datos personales al paciente y a llenar una nueva ficha médica.

La figura N°1 describe cómo el consultorio lleva a cabo esta actividad.

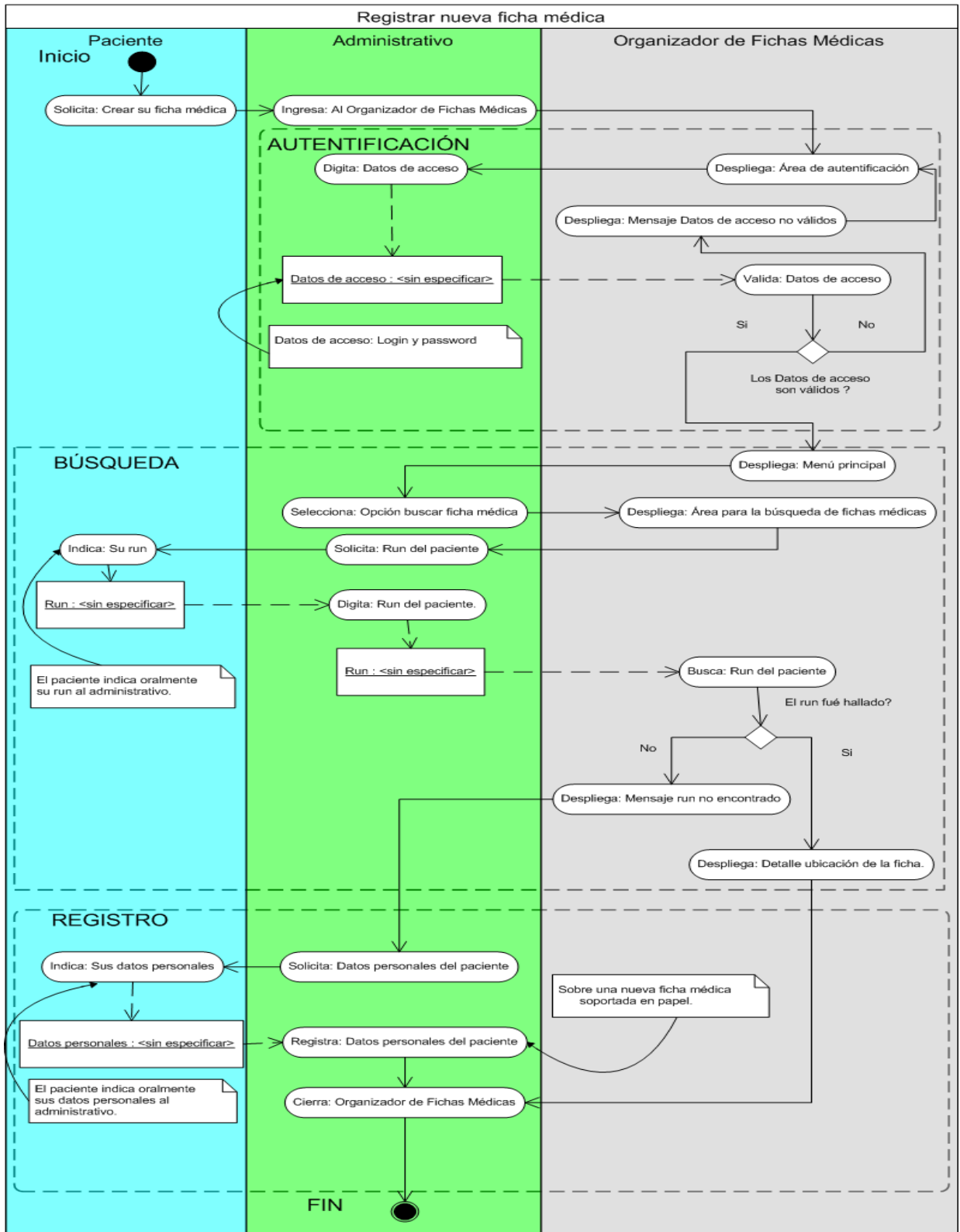


Figura 1, Diagrama de Actividad de registrar nueva ficha médica.

1.2.2. Anexar información a una ficha médica

Anexar información a la ficha médica de un paciente es un procedimiento sólo permitido al personal que directamente lo está atendiendo, se lleva a cabo durante o después de la visita médica, incluye la opinión profesional del personal a cargo, diagnósticos varios, prescripción de medicamentos, entre otros. Por ningún motivo se permite quitar información de la ficha y su manipulación está estrictamente limitada al personal médico.

La figura N°2 describe cómo el consultorio lleva a cabo esta actividad.

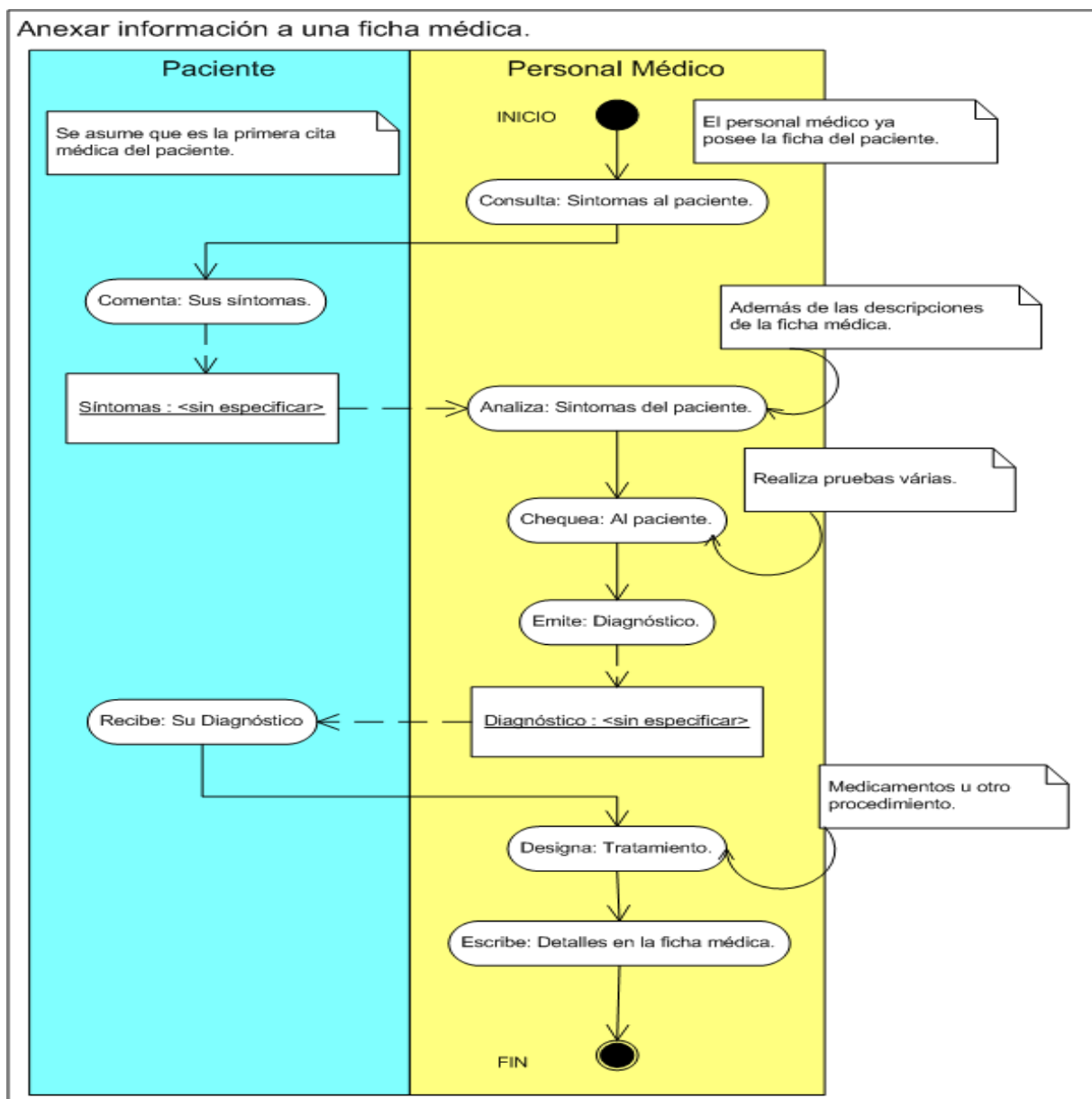


Figura 2, Diagrama de actividad anexar información a una ficha médica.

1.2.3. Conocer las citas asignadas

El consultorio Dr. Federico Puga Borne internamente está organizado en sectores, cada sector es asistido por un médico más una secretaria. Esta última se apoya de un sistema de reservas médicas que permite administrar las citas concernientes al sector y, al inicio de cada jornada, elabora un informe que describe la planificación de horas que se deben cumplir. Por último, el informe es impreso y distribuido al personal médico correspondiente.

La figura Nº 3 describe cómo el consultorio lleva a cabo esta actividad.

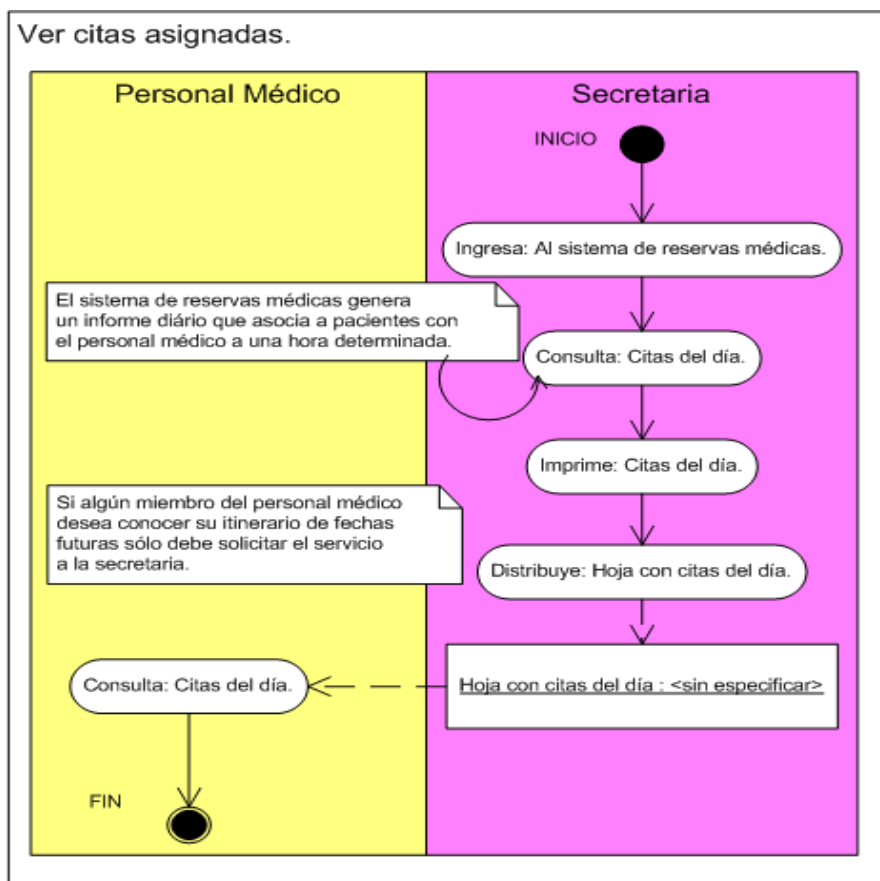


Figura 3, Diagrama de actividad ver citas asignadas.

1.2.4. Ubicar la dirección de los pacientes que serán visitados a domicilio

Este es un aspecto que trasciende a cualquier equipo médico que se desenvuelve en terreno. Ubicar la dirección de un paciente puede convertirse en algo realmente complicado, sobre todo si la ciudad es amplia. En la actualidad, las instituciones médicas que proveen servicios de esta naturaleza han intentado palear el problema apoyándose de mapas callejeros o visitas previas de reconocimiento, los que de igual forma incurren en costos elevados y no aseguran un resultado del todo confiable. Para los del sector público el tema se presenta de un modo más simplificado, esto gracias a la continua iniciativa del gobierno por querer construir un centro de atención médica en cada sector del país, disminuyendo el área de búsqueda.

En el consultorio Dr. Federico Puga Borne esta labor se afronta de la siguiente manera:

En primera instancia el equipo médico a cargo se preocupa de averiguar los datos personales del paciente, específicamente su dirección y un número telefónico al que se pueda contactar. Estos se encuentran registrados en su ficha médica, por ende es estrictamente necesario haberla registrado con anterioridad.

Lo siguiente es establecer contacto telefónico con el paciente para ratificar su dirección y prevenir cualquier inconveniente. Si la llamada no tiene éxito la visita médica se aplaza a una fecha próxima. Caso contrario, se comunica inmediatamente la dirección al conductor de turno quien por medio de variadas fuentes intentará encontrar el sector. Al mismo tiempo el personal médico se encarga de realizar los preparativos que requiera la visita. Algunos factores que apoyan este proceso son las visitas de reconocimiento previo, censos de población y de los sectores asignados a cada institución.

La figura Nº 4 describe cómo el consultorio lleva a cabo esta actividad.

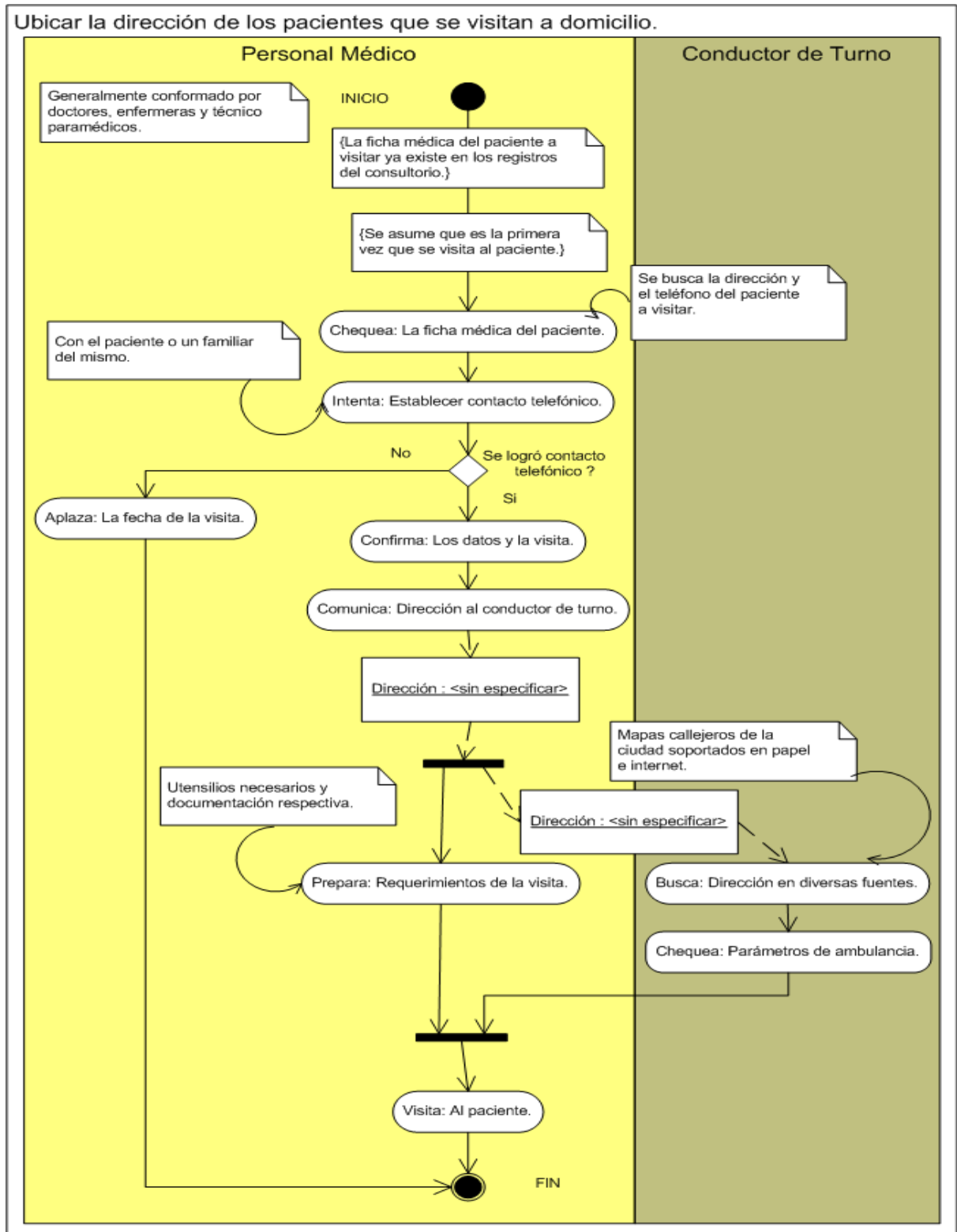


Figura 4, Diagrama de actividad ubicar dirección de pacientes

1.3. DESCRIPCIÓN DEL PROBLEMA

La cantidad de pacientes que a diario requieren atención médica domiciliaria se acrecienta cada vez más. Son diversas las causales que llevan a las personas a optar por esta modalidad que genera un aumento considerable de trabajo para los médicos, a quienes les urge la necesidad de contar con herramientas que les permitan realizar su labor de la misma forma en que lo harían si se encontrasen en su consulta médica.

1.3.1. Situaciones que promueven la atención médica a domicilio.

Existen variadas situaciones que promueven la atención médica a domicilio, las cuales se especifican a continuación:

- Personas que tienen la capacidad de desempeñar actividades de la vida diaria disminuida a causa de algún tipo de inmovilidad, la cual puede ser total o parcial. Estas personas debido a su problema se ven en la necesidad de solicitar atención a sus hogares.
- Personas mayores de edad a las cuales se les hace costosa la movilización en lugares urbanos.
- Personas que por comodidad prefieren llamar a un médico a sus hogares en vez de acercarse a algún centro de salud ya sea pública o privada.
- La gran cantidad de pacientes que deben atender los centros médicos promueve al uso del servicio domiciliario ya que es personalizado y cómodo.

1.3.2. Problemas detectados

Los problemas identificados en el proceso de visitas domiciliarias que realizan los médicos a sus pacientes se centran principalmente en lo que es el registro de horas de atención e ingreso de fichas médicas. Estos conllevan un mal uso del recurso tiempo, poco orden de los registros de pacientes y mayor carga de trabajo de los profesionales.

Las actividades que se realizan y la problemática que estas generan son las que se describen a continuación.

Actividad 1:

El paciente solicita a la secretaria la creación de una ficha médica.

Problema detectado:

-El hecho de que existan fichas que no estén ingresadas en el sistema genera confusión y desorden, ya que la secretaria tendrá que buscar en dos lugares la posible existencia del paciente, incurriendo en gasto de tiempo.

Actividad 2:

La secretaria del médico realiza las reservas de horas de atención en cuadernos escritos manualmente.

Problema detectado:

-Es difícil la reorganización en caso de existir cancelaciones o modificaciones de horas agendadas.

Actividad 3:

La secretaria anota de forma manual los pacientes que el médico debe visitar para luego entregársela a fin de que este realice la consulta domiciliaria.

Problemas detectados:

-Nuevamente se hace difícil la reorganización en caso de existir cancelaciones y modificaciones.

-Al momento de existir alguna cancelación de hora mientras el doctor está realizando la visita médica, este puede no darse por enterado y acudir en vano a una hora agendada, conllevando a gastos innecesarios de recursos y tiempo.

-Es difícil agendar nuevas horas de atención en aquellas que fueron canceladas de un instante para otro.

Actividad 4:

En cada visita domiciliaria el médico debe llenar una ficha y de forma manual las observaciones encontradas, para luego, una vez terminadas todas las visitas, pasar la información de todos los pacientes al sistema.

Problema detectado:

-Se realiza dos veces un trabajo que puede ser hecho solo de una vez, ya que el ingreso al sistema de la ficha debería realizarse en el mismo lugar de la visita, ahorrando tiempo y trabajo.

Actividad 5:

Realizar la visita domiciliaria por parte del médico sin más antecedentes que la dirección del paciente.

Problema detectado:

-El médico podría no saber con exactitud dónde queda la dirección de sus pacientes, lo cual podría provocar que no llegue a destino o llegue con demora.

1.4. OBJETIVOS GENERALES Y ESPECÍFICOS.

Cada uno de los objetivos en esta sección expuestos se han planteado dentro del contexto de la problemática que existe en la atención a domicilio de los pacientes por parte de los médicos tratantes.

1.4.1. Objetivo General

El objetivo general del proyecto es desarrollar una aplicación móvil que permita a los profesionales del rubro de la salud, con labores en terreno, administrar las reservas de atención médica y fichas de los pacientes a su cargo. Lo anterior utilizando las bondades del sistema operativo Android y la gama de dispositivos SmartPhone.

1.4.2. Objetivos específicos

- Diseñar y construir una aplicación nativa en plataforma Android que permita acceder y modificar horas de atención a pacientes y fichas médicas.
- Implementar un mecanismo de localización de direcciones domiciliarias de pacientes.
- Aplicar el concepto de usabilidad al diseño de las interfaces de la aplicación móvil.
- Utilizar los servicios disponibles en la plataforma Android como: localización, multimedia, telefonía, persistencia e interacción.

1.5. METODOLOGÍA A UTILIZAR

El proyecto será desarrollado siguiendo el modelo de trabajo **Incremental**, el cual divide todas las labores en pequeños incrementos funcionales que posteriormente serán evaluados por el usuario. Cada incremento contempla la ejecución de pruebas sobre los módulos que lo conforman, lo que permite asegurar el funcionamiento general del sistema (Larman, 1999).

Por otro lado, al inicio de la etapa de desarrollo se elaborará un **Prototipo** no funcional con las interfaces de usuario (GUI), a fin de evaluar un aspecto clave de toda aplicación móvil, la usabilidad.

La metodología de trabajo será **orientada a objetos** para la confección de los procedimientos y **estructurada** en lo referente al diseño de las interfaces de usuario.

Por cuestiones de organización, se implementará el patrón de diseño Modelo Vista Controlador (MVC) y variadas prácticas como el uso de comentarios y documentación del código. Para el modelado de diagramas se hará uso del lenguaje estándar UML.

1.6. TRABAJOS RELACIONADOS

1.6.1. Trabajos relacionados en la Universidad

En la Universidad del Bio-Bio existen proyectos de título creados por alumnos memoristas de la carrera de Ingeniería de Ejecución en Computación e Informática los cuales se encuentran disponibles en las bibliotecas de la dependencia.

Dichas memorias tienen relación en ciertos aspectos con el presente proyecto y son las que se detallan a continuación:

Nombre de la Memoria: Sistema de Apoyo a la Gestión de Reservas de Horas para el Servicio de Orientación Médica y Estadística (SOME) del Consultorio Violeta Parra Chillán.

Autor : Claudio Alejandro Torres Añasco.

Serie : M (DC) 001.6 T636 2007

Esta memoria plantea la construcción de un sistema informático de apoyo a la gestión del Servicio de Orientación Médica y Estadística (SOME) en el Consultorio Violeta Parra de Chillán. Entre sus principales funciones se encuentra el mantener una base de datos con los profesionales de la salud, pacientes y las reservas que estos últimos realizan, de acuerdo a la prestación médica que requieran.

Nombre de la Memoria: Implementación de juego 2D para dispositivos móviles utilizando J2ME.

Autor : Carolina Cecilia González Pavéz.

Serie : M (DC) 001.6 G589 2008

Es una aplicación creada para dispositivos móviles destinada al área de la entretención, haciendo uso de tecnología más antigua y con menos servicios de los que puede ofrecer uno que cuente con plataforma Android.

En este proyecto se destaca, por sobre el juego, el uso de la arquitectura J2ME y sus componentes.

1.6.2. Trabajos relacionados fuera del ámbito académico.

Nombre de la aplicación: Gestión de Citas Médicas.

Página Web: <https://www.gestioncitasmedicas.com/login.php>

Creador: SULIME

Es una aplicación de pago creada para la gestión de citas médicas y es soportada en un servidor Web, el que puede ser accedido mediante un computador o por teléfonos celulares que tengan conexión a Internet.

Permite la gestión de citas de uno o varios profesionales, de una o varias especialidades, la recogida de datos básicos de pacientes y visualización de agenda diaria resumida.

Si bien no es una aplicación contenida en un teléfono celular, como lo es el caso de este proyecto, la aplicación sigue la línea de una agenda que permite llevar el control de las citas e información del paciente.

Nombre de la aplicación: Agenda Médica DEMOSYS

Página Web: <http://www.demosys.com/demosys/portal/index.php?>

Creador: DEMOSYS electronics and systems engineering

Es una agenda gratuita creada para ayudar a médicos en general, permite la administración de citas de pacientes, manejo de citas en formato de notas autoadhesivas, contiene un calendario en donde se pueden ver los días con anotaciones pendientes y ver el detalle de los pacientes a atender en el día.

Es una agenda de escritorio y, al contrario de la descrita anteriormente, esta no permite la conexión desde otros lugares de trabajo, debe ser utilizada solo en el computador instalado, presentando una desventaja ya que obliga al médico a permanecer en la central de trabajo para revisar sus citas.

Capítulo II

TECNOLOGÍA PARA DISPOSITIVOS MÓVILES

La telefonía celular ha crecido a pasos agigantados estos últimos años. Son miles las personas que cuentan con dispositivos móviles y hacen uso de los servicios ofrecidos por estos.

Para funcionar correctamente los dispositivos deben encontrarse insertos en un ecosistema tecnológico, el cual se compone de servicios, operadores, redes y plataformas.

En este capítulo se explica dicho ecosistema y de qué forma colaboran para ofrecer un servicio de calidad.

2.1. TELEFONÍA MÓVIL ACTUAL EN CHILE

La telefonía móvil se implantó en Chile hace más de veinticinco años, en el gobierno de Augusto Pinochet, quienes fueron los que licitaron las frecuencias necesarias para abastecer el territorio chileno.

En sus comienzos el segmento de la población a la que estaba dirigida el producto era la clase alta, debido a que existían pocas empresas que fabricaban teléfonos celulares, los altos costos existentes por el hecho de tener que pagar por llamada emitida y recibida, el cobro de roaming nacional en caso de encontrarse en un área ajena a la que cubría la empresa prestadora de servicios y por último el alto costo de los terminales.

En el año 1988 existían tres empresas que proveían el servicio de comunicación celular, estas empresas eran CTC Celular, Telecom Celular y VTR Celular.

Por los años noventa CTC Celular adquiere VTR Celular naciendo Startel. Esta unión logra posicionar a esta última como la única empresa de telefonía celular chilena que poseía cobertura en todo el territorio nacional.

A mediados de los noventa el segmento de la población que poseía un teléfono celular había crecido considerablemente puesto que ahora era más barato obtener uno que en sus comienzos y las empresas de servicios móviles disminuyeron los costos de sus servicios.

Hoy en día más de 15.800.000 chilenos cuentan en su poder con un teléfono celular, siendo Movistar, ENTEL PCS y Claro los operadores más masificados en el territorio nacional. Cabe considerar que la cobertura de la telefonía móvil en Chile es del 76% contra un 21% de la telefonía fija esto según investigaciones del INE (Instituto Nacional de Estadísticas). Siendo Chile considerado uno de los países sudamericanos y a nivel americano con el mayor nivel de penetración en telefonía móvil.

Capítulo II, Tecnología para dispositivos móviles

Según la Subsecretaría de Telecomunicaciones (Subtel) el crecimiento anual de la telefonía celular en Chile de diciembre del 2003 a diciembre del 2009 ha sido la que se muestra en la figura N°5.

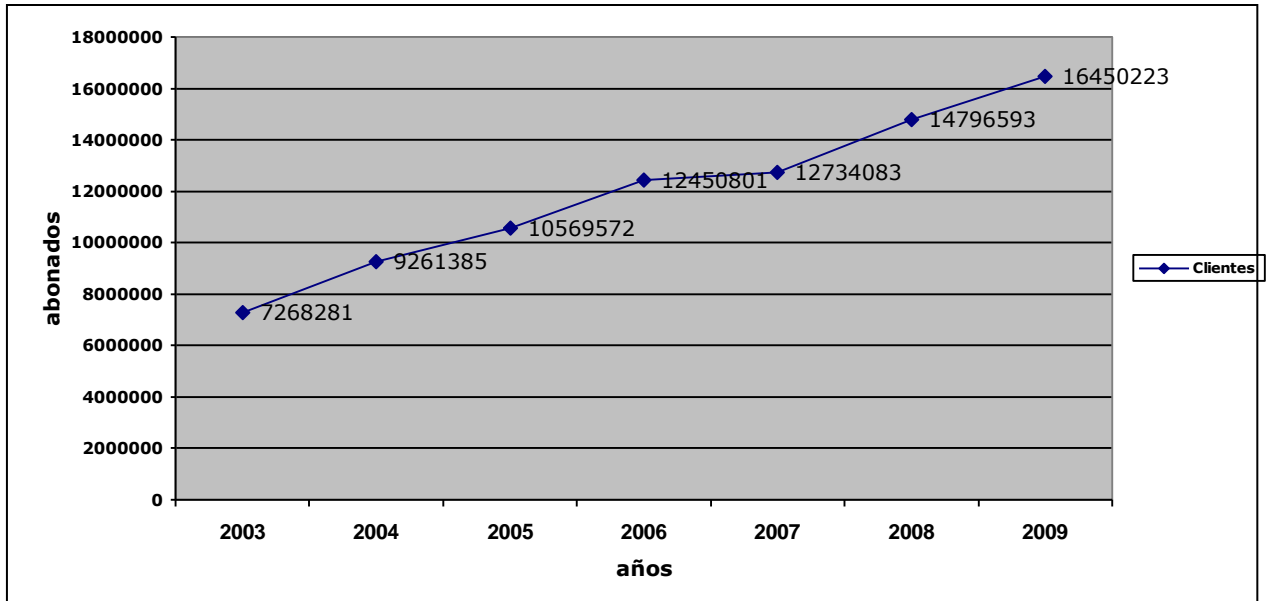


Figura 5, Gráfico de crecimiento anual en telefonía móvil

Las compañías de telefonía móvil ofrecen servicios con prepagado y plan, estos tienen distintos porcentajes de usuarios dependiendo de las preferencias, a continuación en la figura N°6 se muestra el detalle de estos por cada compañía en el año 2009 según la Subtel.

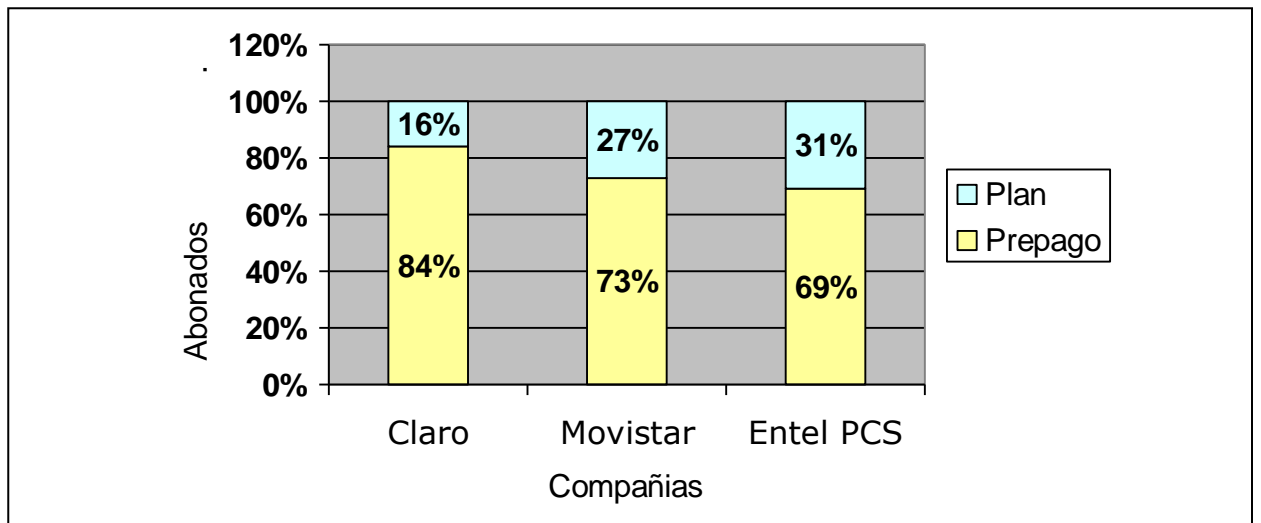


Figura 6, Gráfico de preferencias de servicio

2.2. ECOSISTEMA MÓVIL

Se conoce como ecosistema móvil al conjunto de organismos compuesto por operadores, redes, fabricantes de dispositivos, plataformas que dependen unos de otros para hacer más lucrativos sus negocios y ofrecer servicios de calidad.

Este ecosistema se organiza en capas y se ordenan de la forma que se muestra en la figura N° 7:

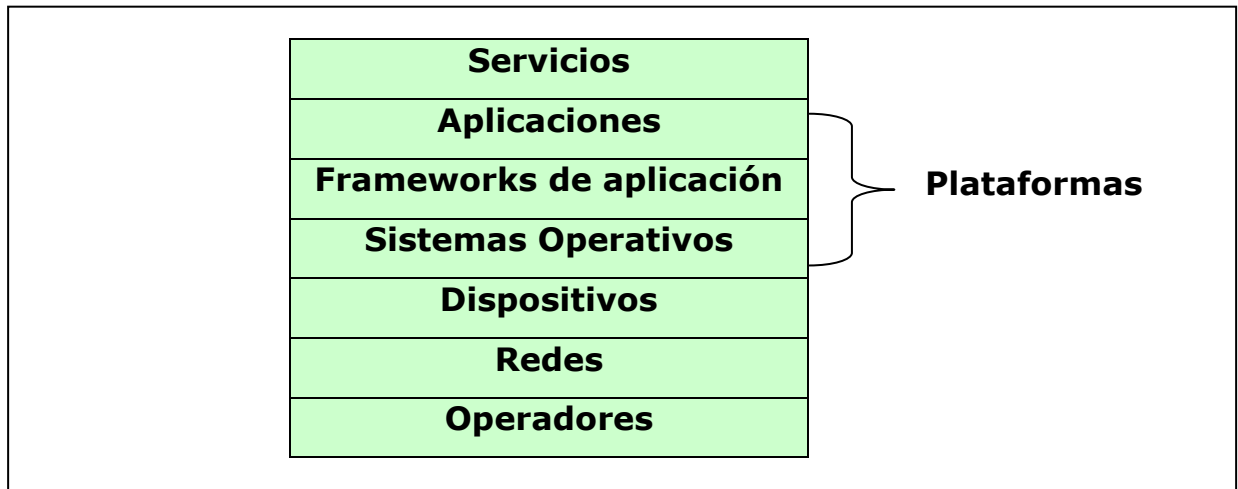


Figura 7, Capas ecosistema móvil.

Cada capa depende de la anterior, pero no necesariamente todos los servicios móviles utilizan todas las capas.

2.2.1 Operadores de telefonía móvil

Los operadores de telefonía móvil son compañías telefónicas que ofrecen servicios de telefonía a clientes de teléfonos móviles. Esencialmente son los que logran que todo el ecosistema funcione encargándose de crear y mantener servicios de tecnología inalámbrica a través de una red celular confiable. En Chile, actualmente existen tres operadores de gran importancia las cuales son MOVISTAR, ENTEL PCS y CLARO.

ENTEL PCS es una operadora de servicios telefónicos para dispositivos móviles, es la segunda más importante en el país posee más de 6,2 millones de abonados. Se ha

Capítulo II, Tecnología para dispositivos móviles

destacado por ser pionera en servicios en Chile ya que el 2006 lanzó la primera red 3.5G UMTS/HSDPA la que llega a todas las capitales regionales y ciudades mayores del país, fue la primera en utilizar la tecnología GSM la que actualmente llega a la totalidad de las áreas urbanas y periurbanas del país, así como también a las principales carreteras y a diversas zonas rurales; siendo la única operadora que presta servicios en localidades aisladas como Puerto Williams, Isla de Pascua y las bases en la Antártica Chilena.

Actualmente ENTEL PCS realiza pruebas de banda ancha móvil de 4G LTE (uno de los 4 operadores en el mundo), gracias a un acuerdo con Ericsson. Se estima que la red LTE estaría disponible en Chile para 2011.

Movistar es la operadora de servicios telefónicos móvil más importante del país con 7 millones de clientes. Fue la primera en recibir el año 2008 el Premio a la Innovación de la consultora Frost & Sullivan, es considerada la 5° mejor empresa para trabajar en el país según el ranking Great Place to Work.

Claro operadora de servicios móviles con más de 3,3 millones de clientes lo que la sitúa como la tercera compañía en cuanto a cobertura y cantidad de abonados, cuenta con una amplia cobertura a nivel nacional.

Durante 2007 y 2008, Claro Chile invirtió más de US\$ 500 millones en expandir su cobertura y mejorar su red, totalizando más de 1.700 antenas dentro del territorio e ofreciendo cobertura 3G desde Arica hasta Punta Arenas.

Según estadísticas proporcionadas por la Subtel, las posiciones de las empresas antes descritas en el mercado de la telefonía móvil es la que se muestra en la figura N° 8.

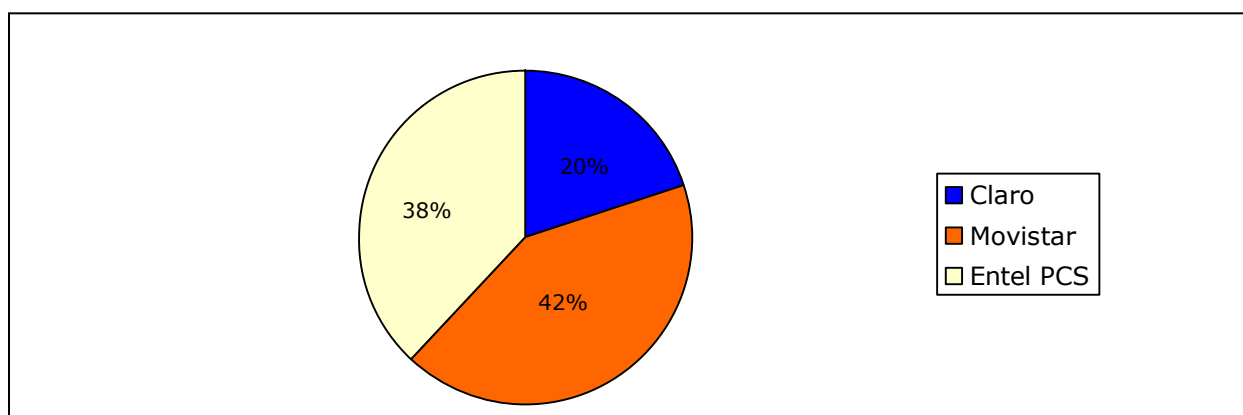


Figura 8, Posición en el mercado de operadores móvil

2.2.2. Redes

2.2.2.1 Redes de telefonía móvil (PLMN)

La red de Telefonía móvil consiste en un sistema telefónico en el que, mediante la combinación de una red de estaciones transmisoras-receptoras de radio (estaciones base) y una serie de centrales telefónicas de conmutación, se posibilita la comunicación entre terminales telefónicos móviles. Son los operadores de telefonía móvil los que mantienen y utilizan estas redes.

La figura Nº 9 muestra un esquema simplificado de lo que es la red de telefonía móvil.

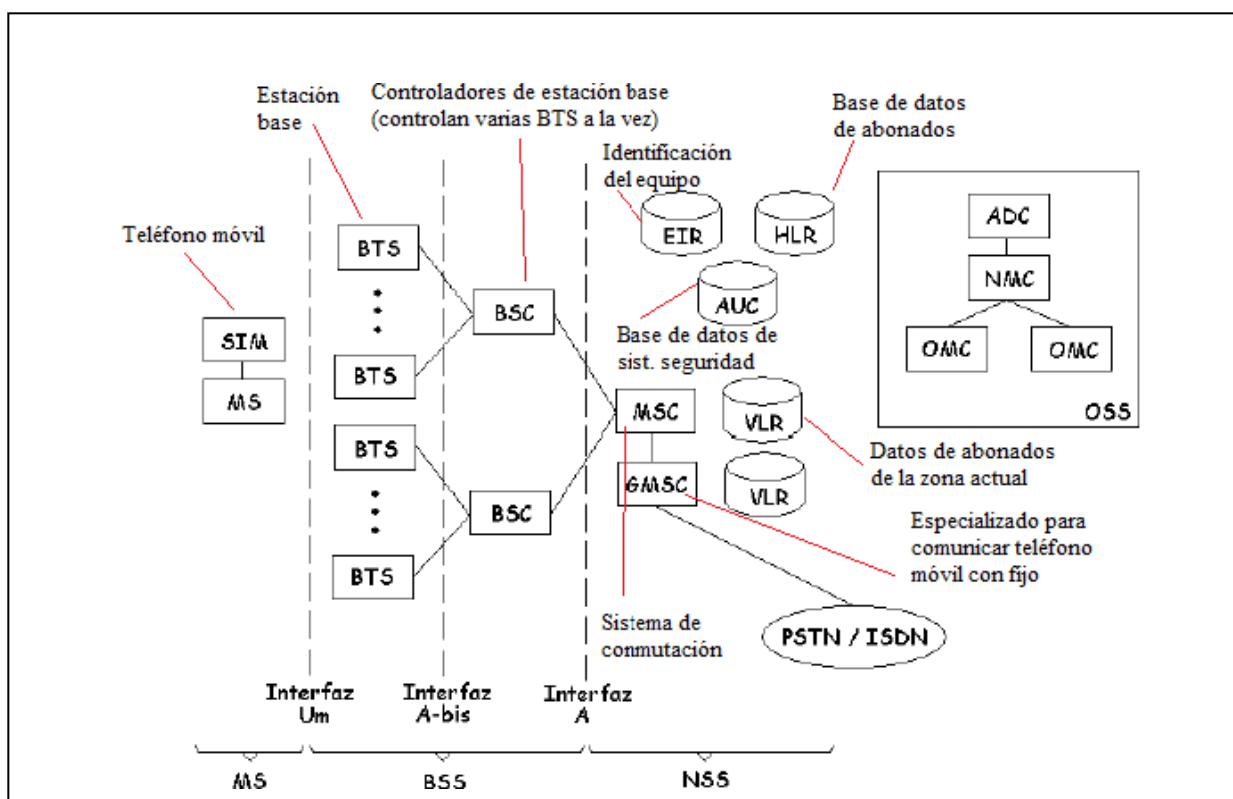


Figura 9, Esquema de Red de telefonía móvil
(Breve introducción a la red de telefonía móvil, 2010)

Como se puede ver en la imagen Nº9, hay tres siglas en la parte inferior la MS, BSS y NSS. Estas son las siglas correspondientes a estación móvil (*Mobile Station*), subsistema de estación base (*Base Station Subsystem*) y subsistema de red y conmutación (*Network &*

Capítulo II, Tecnología para dispositivos móviles

Switching Subsystem), respectivamente, las cuales analizaremos con más detalle a continuación.

Estación móvil (MS)

Es la parte referida al teléfono móvil en sí, en esta sección se encuentran:

- **IMEI:** La identidad internacional de equipo móvil, la que es la identificación del dispositivo.
- **Tarjeta SIM:** Es la tarjeta desmontable que está en el teléfono móvil y que contiene el número IMSI (identidad internacional de abonado a móvil). Las SIM almacenan de forma segura la clave de servicio del suscriptor usada para identificarse ante la red, de forma que sea posible cambiar la línea de un terminal a otro simplemente cambiando la tarjeta.

Subsistema de estación base (BBS)

Aquí hay dos elementos importantes: la estación base (BTS) y el controlador de estación base (BSC)

- **BTS:** La estación base contiene computador, un transmisor/receptor, procesadores de señal y medidas de canal. Todas las BTS se conectan a una oficina de conmutación de telefonía móvil.
- **BSC:** Se encarga de controlar las estaciones base a su cargo y de establecer el canal de radio, de los saltos de frecuencia, traspaso dentro del controlador (pasa de una BTS a otra si existen problemas con la señal de un móvil tras las medidas de canal), la configuración de las estaciones, la supervisión del enlace radio y el control de potencia (*Breve introducción a la red de telefonía móvil, 2010*).

Subsistema de red y conmutación

Conformada por los siguientes dispositivos:

- **MSC (Mobile Switch Center):** Es el centro de conmutación móvil, encargado del encaminamiento de las llamadas y de gestionar los abonados móviles (registro,

autenticación, traspaso) en colaboración con otras entidades de la propia red. También realiza la conexión a la red fija (de hecho, tiene una especialización al respecto, llamada GMSC).

- **HLR:** El Registro de Localización Base es una base de datos relativa al abonado e información de localización. Hay un HLR por cada red GSM (puede implementarse distribuido).
- **VLR (Visitor Locating Register):** El registro de Localización del Visitante es una base de datos de los abonados de la zona, aunque puede incluir varias áreas de localización. Y por norma general, MSC y VLR van juntos.
- **EIR (Equipment Identity Register):** El Registro de Identidad del equipo es una base de datos de los equipos móviles, conteniendo los IMEI válidos e inválidos. Es bastante útil en caso de móviles robados, aunque en otros países, al no estar en la misma red, el teléfono bloqueado queda "desbloqueado"; sin embargo, existen acuerdos entre países para evitarlo.
- **AUC (Authentication Center):** El Centro de Autenticación es una base de datos protegida. Contiene los números secretos para autenticación (que también está guardado en la tarjeta del teléfono), además de generar tripletas para cifrado. Generalmente está asociado a HLR.

2.2.2.2 Sistema GSM (Global System for Mobile communications)

El sistema GSM es el estándar de telefonía móvil más usado en el mundo y fue creado para poder comunicarse en toda Europa con telefonía móvil usando el mismo sistema. GSM, además, trabaja con un sistema de celdas que se dividen en cinco tipos:

- **Macro celdas:** Se pueden tomar como las celdas donde se instala la antena de la estación base en un mástil o en un edificio, por encima de la altura media de tejado.
- **Micro celdas:** La altura de las antenas está por debajo de la altura media de tejado y son las más usadas en áreas urbanas.
- **Pico celdas:** Pequeñas celdas con una cobertura de unas pocas docenas de metros, usadas normalmente en interiores.

- **Femto celdas:** Celdas diseñadas para uso residencial o en pequeños negocios.
- **Celdas paraguas:** Cubren regiones "en sombra" (donde no llega la señal) de las celdas más pequeñas y cubre los huecos de cobertura entre éstas.

Los sistemas de telefonía móvil modernos utilizan celdas debido a que las frecuencias de radio son recursos limitados y compartidos. Las celdas y los dispositivos cambian de frecuencia bajo el control de computadoras y usan transmisores de baja potencia, así el número limitado de frecuencias puede ser utilizado por muchas llamadas con menos interferencia.

Se llaman celdas porque su área de cobertura, genera un espacio parecido al de una celda (como se puede ver en la figura N°10), de aquí viene el nombre de "celulares".

Cuando el teléfono móvil sobrepasa la cobertura de una celda ésta traspasa el mando a otra celada para que esta se haga cargo.

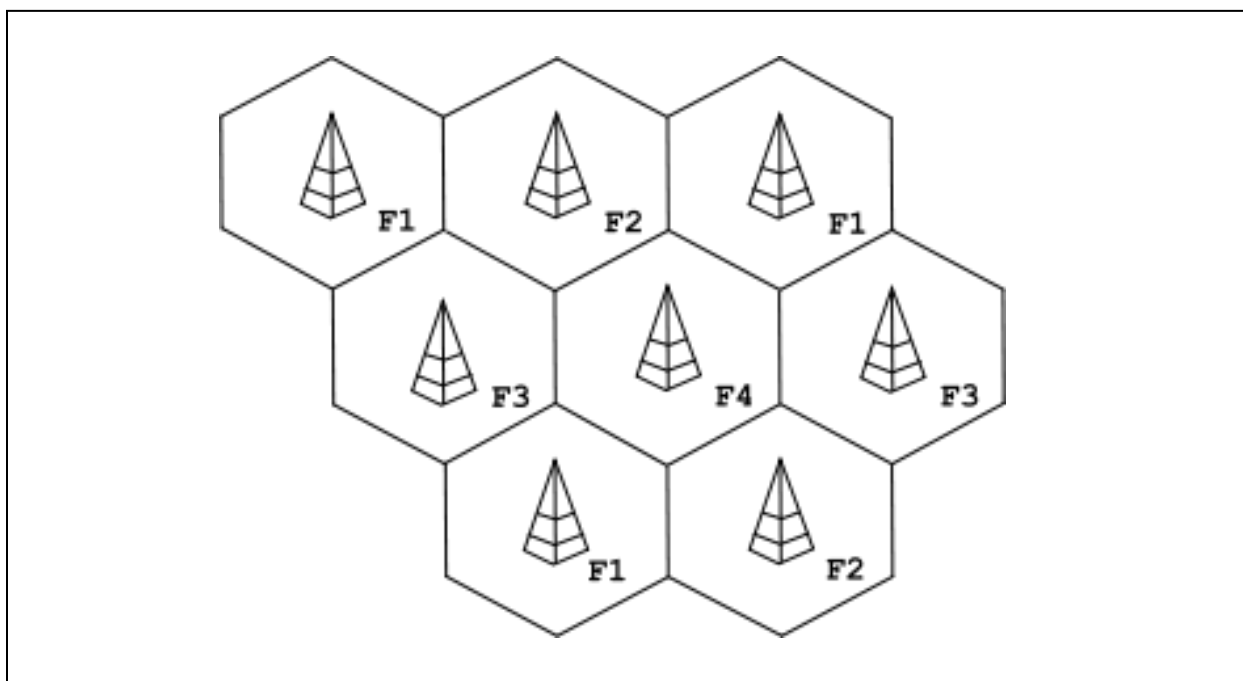


Figura 10, Disposición de celdas celulares (Red de celdas, 2010).

2.2.2.3. Tecnologías de telefonía móvil

Las tecnologías que a continuación se detallan implican redes de telecomunicaciones, velocidades de transmisión y distintos equipos móviles

Primera generación 1G: Es la abreviación para la tecnología de primera generación en la telefonía móvil, hizo su aparición en 1979, se caracterizó por ser analógica y estrictamente para voz. Era bastante básica ya que la calidad de los enlaces de voz era muy baja, baja velocidad, la transferencia entre celdas era muy imprecisa, tenían baja capacidad y la seguridad no existía.

Segunda generación 2G: Apareció alrededor de los años 90 y, a diferencia de la primera generación, esta es digital.

El sistema 2G utiliza protocolos de codificación más sofisticados y son los sistemas de telefonía celular usados en la actualidad.

Los protocolos empleados en los sistemas 2G soportan velocidades de información más altas para voz, pero limitados en comunicaciones de datos. Se pueden ofrecer servicios auxiliares tales como datos, fax y SMS (Mensajería de texto).

La generación 2.5G: Posee características extendidas para ofrecer capacidades adicionales que los sistemas 2G, tales como GPRS (General Packet Radio System), HSCSD (High Speed Circuit Switched Data), EDGE (Enhanced Data Rates for Global Evolution), entre otros.

El término 2.5G no está definido oficialmente, fue inventado solo con fines comerciales.

La tercera generación 3G: Es tipificada por la convergencia de la voz y datos con acceso inalámbrico a Internet, aplicaciones multimedia, altas transmisiones de datos, e-mail, mensajería instantánea.

Los protocolos empleados en los sistemas 3G soportan más altas velocidades de información enfocados para aplicaciones mas allá de la voz tales como audio (MP3), video en movimiento, video conferencia y acceso rápido a Internet.

Es en esta generación en donde se encuentra el dispositivo móvil HTC Magic descrito en el punto 2.2.3 de este capítulo.

La cuarta generación 4G : Aún no está bien definido en qué consiste la tecnología 4G, hasta el momento cuando se habla de este término se hace referencia a la mejora de las prestaciones existentes en la 3G.

Una de las principales ventajas que se estima que tenga la 4G son velocidades de transmisión 10 veces más rápido que la actual tecnología.

2.2.3. Dispositivos Móviles

Son dispositivos móviles aquellos aparatos suficientemente pequeños para ser transportados y utilizados por personas independiente de su ubicación.

Las características principales de estos dispositivos son las que se enumeran a continuación:

1. Su tamaño es reducido, para facilidad de transporte.
2. Ofrecen conexión a Internet, la cual puede ser permanente o intermitente.
3. Poseen una memoria limitada.
4. Normalmente se asocian al uso individual de una persona, tanto en posesión como en operación, el cual puede adaptarlos a su gusto.
5. Diseñados específicamente para una función, pero que pueden llevar a cabo otras más generales.

Entran en la categoría de dispositivo móvil los siguientes:

1. **PDA:** Es un computador de mano, el cual originalmente era diseñado como agenda electrónica. Hoy en día ofrece variados servicios similares a los de un PC portátil como por ejemplo ver películas, crear documentos, usar correo electrónico, navegar por Internet, etc.

2. **Computador portátil:** es una computadora personal móvil, son capaces de realizar la mayor parte de las tareas que realizan las computadoras de escritorio, con la ventaja de que son más pequeñas, más livianas y tienen la capacidad de operar por un período determinado sin estar conectadas a la electricidad.
3. **Teléfono inteligente (SmartPhone):** Es un teléfono parecido a un PC portátil pero más limitado, permite conexiones a Internet y la instalación de aplicaciones para incrementar el procesamiento de datos y la conectividad.

Es en esta última categoría en donde se encuentra el teléfono móvil que se usará para el desarrollo de este proyecto. Este Smartphone, de la marca HTC modelo Magic es un teléfono móvil basado en el nuevo sistema operativo, Android™ y corresponde al que se muestra en la figura N° 11:



Figura 11, HTC Magic

2.2.3.1. Características principales HTC Magic

El HTC Magic posee distintas funcionalidades y ha sido creado para entregar la mayor satisfacción al usuario, se caracteriza principalmente por:

Hardware: Estéticamente es atractivo son delgados y muy livianos (118,5 gramos con batería), cuenta con los botones suficientes para poder navegar por el interfaz, aunque el fuerte de este terminal y su sistema operativo es que se maneja de manera Touch. Pese a ello cuenta con trackball multifunción para poder desplazarse por todo el menú.

Posee una memoria ROM de 512MB y una RAM de 192MB.

Pantalla: Se trata de una pantalla táctil TFT-LCD de 3,2 pulgadas más pequeña que las pantallas del iPhone, Storm y similares. Posee una resolución HVGA (320 x 480) y un brillo sobresaliente.

Batería: Cuenta con una batería recargable de ion de litio con una capacidad de 1.340 mAh. La duración de la batería es la siguiente:

Tiempo de conversación:

- Hasta 400 minutos en WCDMA
- Hasta 450 minutos en GSM

Tiempo de espera:

- Hasta 660 horas en WCDMA
- Hasta 420 horas en GSM

Todas estas cifras están sujetas a la red y al uso que se haga del teléfono debido a que tareas como el uso de WiFi y GPS aumentan el gasto de batería.

Cámara fotográfica: Posee una cámara de 3,2 megapíxeles con enfoque automático, cumple los estándares de los móviles con los que compite.

GPS: Abreviatura de Sistema de posicionamiento global, el cual es un chip integrado en el teléfono que permite determinar en todo el mundo la posición de un objeto, una persona, un vehículo o una nave, con unos pocos metros de imprecisión.

Capítulo II, Tecnología para dispositivos móviles

Conectividad: Cuenta con Bluetooth® 2.0 para la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz., con Wi-Fi®: IEEE 802.11 b/g para la conexión a Internet y HTC ExtUSB™ el cual es un mini-USB 2.0 y conector de audio en uno.

Acelerómetro: Es un tipo de sensor que posee el teléfono, el cual permite medir la reacción de un objeto sometido a una fuerza, evaluando la dirección y la variación de la velocidad. Es decir, es un dispositivo capaz de convertir ciertos gestos en una señal eléctrica que puede ser o no interpretada por el sistema.

Brújula digital: Permite la orientación.

2.2.4. Plataformas

Con el avance de las tecnologías, hoy en día los teléfonos móviles se han vuelto inteligentes gracias a las plataformas diseñados especialmente para ellos.

2.2.4.1. Sistemas Operativos

Los sistemas operativos permiten el control de un dispositivo móvil, de la misma manera que lo hace Windows o Linux en un computador.

La principal diferencia que existe es que el sistema operativo móvil es mucho más simple y está orientado a la conectividad inalámbrica. Están divididos en capas, las cuales se especifican a continuación:

Kernel: Núcleo que proporciona el acceso a los distintos elementos del hardware del dispositivo. Es el encargado de gestionar recursos, a través de servicios de llamada al sistema, se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo, ofrece distintos servicios a las capas superiores como son los drivers para el hardware.

Capítulo II, Tecnología para dispositivos móviles

Middleware: Conjunto de módulos que hacen posible la propia existencia de aplicaciones para móviles. Es totalmente transparente para el usuario y ofrece servicios claves como el motor de mensajería y comunicaciones, codecs multimedia, intérpretes de páginas Web, gestión del dispositivo y seguridad.

Entorno de ejecución de aplicaciones: Consiste en un gestor de aplicaciones y un conjunto de interfaces programables.

Interfaz de usuario: Facilitan la interacción con el usuario y el diseño de la presentación visual de la aplicación. Los servicios que incluye son el de componentes gráficos (botones, pantallas, listas, etc.) y el del marco de interacción.

Existen variados sistemas operativos disponibles para dispositivo móviles dependiendo de la empresa que los construye o para qué tipo de dispositivo, generalmente los más usados son:

Symbian: Es un sistema operativo para dispositivos móviles más masificado ocupa el 65% del mercado. Fue producto de la alianza de varias empresas de telefonía móvil, entre las que se encuentran Nokia, Sony Ericsson, LG, Motorola, Mitsubishi Electric, Panasonic, Sharp entre otras. Sus orígenes provienen de su antepasado EPOC32, utilizado en PDA's y Handhelds de PSION.

Symbian tiene amplia variedad de aplicaciones disponibles para todo tipo de teléfonos móviles.

BlackBerry OS: Sistema operativo creado por la RIM (Research In Motion). Es multitarea, permite a los usuarios seguir conectados al correo electrónico y a muchas aplicaciones corporativas en donde se encuentren. Este sistema soporta desarrollo de aplicaciones Java para móviles con los perfiles MIDP 1.0 y desde la versión 4 de BlackBerry en MIDP 2.0.

Actualmente BlackBerry OS cuenta con un 11% del mercado.

Windows Mobile: Windows Mobile es un sistema operativo escrito desde 0 y que hace uso de algunas convenciones de la interfaz de usuario del Windows de siempre. Cuenta con un conjunto de aplicaciones básicas utilizando las API de Microsoft Windows. Está diseñado para ser similar a las versiones de escritorio de Windows estéticamente. Además, existe una gran

oferta de software de terceros disponible para Windows Mobile, la cual se puede adquirir a través de Windows Marketplace for Mobile.

Windows Mobile cuenta con el 12% del mercado

iPhone OS: Sistema operativo que utiliza el iPod touch, iPhone e iPad, diseñado por 175 ingenieros de Apple. Está basado en una variante del Mach kernel que se encuentra en Mac OS X. El iPhone OS incluye el componente de software "Animación Core" de Mac OS X v10.5 que, junto con el hardware de 3D, PowerVR MBX, es responsable de las animaciones usadas en el interfaz de usuario.

Android: Es un sistema operativo orientado a dispositivos móviles basado en una versión modificada del núcleo Linux. Inicialmente fue desarrollado por Android Inc., compañía que fue comprada después por Google, y en la actualidad lo desarrollan los miembros de la Open Handset Alliance (liderada por Google).

La presentación de la plataforma Android se realizó el 5 de noviembre de 2007 junto con la fundación Open Handset Alliance, un consorcio de 48 compañías de hardware, software y telecomunicaciones comprometidas con la promoción de estándares abiertos para dispositivos móviles. Esta plataforma permite el desarrollo de aplicaciones por terceros a través del SDK, proporcionada por Google, y mediante el lenguaje de programación Java.

2.2.4.1. Framework de aplicaciones

El framework puede ser definido como un conjunto de clases y las colaboraciones que se establecen entre ellas, para proporcionar un diseño abstracto para las soluciones de un conjunto de problemas.

El framework captura las decisiones de diseño comunes a un tipo de aplicación, estableciendo un modelo común a todas ellas, asignando responsabilidades y estableciendo colaboraciones entre las clases que forman el modelo.

En específico un framework de aplicación encapsula una capa de funcionalidad horizontal que puede ser aplicada en la construcción de una gran variedad de programas.

Capítulo II, Tecnología para dispositivos móviles

A menudo el desarrollador puede acceder la primera capa del Framework de aplicación o API. La primera capa tiene control de la elección del marco de aplicación.

Los frameworks de aplicación a menudo se ejecutan en la capa superior de los sistemas operativos, compartiendo servicios básicos tales como comunicaciones, mensajes, gráficos, localización, seguridad, autenticación y muchos otros.

Todas las plataformas dependiendo del tipo, ya sea para iPhone, BlackBerry, Android u otros tienen distintos Framework de aplicación.

Algunos más conocidos son:

Framework S60:

- Es multitarea.
- Protección de memoria que implementa un entorno de ejecución robusto y seguro.
- Software de gestión de energía, ahorra energía en dispositivos con pilas
- Bluetooth y conectividad TCP/IP.
- Soporte de idiomas asiáticos.
- Interfaz multitouch.
- Se pueden ejecutar aplicaciones Java ME.
- Posee mejoras para apoyar a las empresas.
- Amplia gama de posibilidades para desarrollar aplicaciones S60, utilizando diferentes lenguajes de programación.

Framework RIM Blackberry

- Es multitarea.
- Proporciona una protección de memoria y la ejecución de código de confianza.
- Se puede ejecutar aplicaciones Java ME.
- Soporte para interfaz táctil.
- Maneja múltiples cuentas de correo electrónico de Exchange, con soporte para POP3 y SMTP, los mensajes de correo electrónico pueden tener archivos adjuntos.
- Browser soporta varios tipos de medios.
- Permite sesiones de cifrado a través de BlackBerry Enterprise Server.

- Tiene herramientas de desarrollo para escribir aplicaciones Java ME para teléfonos inteligentes BlackBerry.

Framework iPhone

- Posee el navegador Web con la mejor experiencia de los smartphones, Safari.
- Es capaz de descargar, seleccionar y reproducir música.
- Es una plataforma de aplicaciones, capaz de ejecutar programas utilitarios de diversa índole.
- Cuenta con el apoyo de la empresa, como la sincronización con Exchange, y la posibilidad de establecer una conexión VPN.
- API basados en la localización mediante GPS, el dispositivo proporciona información de posición, valioso para la creación de nuevas formas de presencia local y las aplicaciones de redes sociales.
- API para permitir el acceso a los datos del acelerómetro, permitiendo el desarrollo de novedosos juegos.

2.2.5. Servicios

Son variados los servicios móviles que existen los cuales dependen de la compañía operadora que los ofrezca y las capacidades que tenga el teléfono móvil, algunas de estas son:

- Acceso a Internet
- Envío de un mensaje de texto
- Video Call
- Envío de mensajes multimedia
- Localización geográfica del equipo

Capítulo III

ANDROID

Como se ha mencionado en el capítulo anterior, Android es un sistema operativo para dispositivos móviles basado en Linux, el cual fue desarrollado hace pocos años atrás por Android Inc. que hoy en día pertenece a Google.

En este capítulo se analizará la plataforma para dispositivos móviles Android, a fin de conocer detalladamente cómo está estructurada su pila tecnológica, un poco de su historia, y ver algunas aplicaciones sencillas. Cabe recordar que esta plataforma será utilizada en este proyecto.

3.1. HISTORIA DE ANDROID

Android es una plataforma móvil desarrollada por Android Inc. una pequeña compañía cuya finalidad era el desarrollo de aplicaciones para dispositivos móviles, la cual fue comprada por Google en Julio del 2005 y fue justamente en esa fecha que comenzaron a nacer los primeros rumores sobre la posible incursión de Google en el mundo de la telefonía móvil.

Google contrató a los fundadores de Android Inc, entre ellos Andy Rubin quien sería luego el director de la división de plataformas móviles de Google.

Si bien la marca Android era desconocida en esos años, el grupo de fundadores tenía gran experiencia en plataformas Web, telecomunicaciones y aplicaciones móviles.

La presentación oficial de Android se realizó el 5 de Noviembre de 2007 junto con la creación de la Open Handset Alliance, una alianza que desarrolla normas abiertas para dispositivos móviles, lo que permite contar con estándares establecidos para diseño, bibliotecas y herramientas para el desarrollo. Estas están disponibles de forma gratuita y para cualquier tipo de usuario.

Cinco días después de la presentación de Android, Google lanza un Software denominado "Development Kit o SDK de Google", que incluía un emulador de Android que permite ir probando aplicaciones creadas por los usuarios representado gráficamente la interfaz de un teléfono móvil.

Cabe destacar que antes de presentar el nuevo sistema operativo, Google se aseguro de contar con compañías que comercializaran terminales impulsados por esta plataforma, fue por esto que firmó un acuerdo con 34 compañías entre las que se encontraba Samsung, HTC, Qualcomm, Motorola, Telefónica y T-Mobile, las cuales se comprometían a utilizar Android en sus dispositivos móviles como sistema operativo.

La primera versión de un teléfono móvil con Android fue el G1 T-Mobile G1/HTC Dream, anunciado el 23 de Septiembre de 2008 y que se lanzó al mercado estadounidense el 22 de Octubre del mismo año. Otro modelo es el HTC Magic anunciado el 18 de febrero de 2009. Se estima que se unan nuevos teléfonos con sistema operativo Android como Lenovo, Sony Ericsson, Motorola, LG y Samsung.

Ahora bien, si se analiza la ventaja que posee Android respecto a sus competidores como iPhone S.O. o Windows Mobile se destaca el hecho que su kernel basado en Linux puede ser adaptado en casi cualquier dispositivo sin incurrir en licencias para uso.

3.2. CARACTERÍSTICAS TÉCNICAS DE ANDROID

Android, por el hecho de haber sido creado en base al Kernel Linux, es gratuito lo que permite realizar variadas actividades con él, sin incurrir en costos de licencias. Entre las bondades que ofrece se encuentran:

- Crear aplicaciones utilizando una amplia gama de útiles bibliotecas y herramientas que pueden ser utilizadas para construir aplicaciones variadas.
- Está construido en código abierto lo que permite ser ampliado para incorporar nuevas tecnologías si los usuarios así lo desean.
- Ofrece una máquina virtual personalizada que ha sido diseñada para optimizar la memoria y los recursos de hardware en un entorno móvil.
- La plataforma Android continuará evolucionando a medida que la comunidad de desarrolladores trabajen para crear aplicaciones móviles.
- Permite ofrecer a los usuarios un amplio espectro de aplicaciones y servicios ya que cualquier persona puede desarrollarlas libremente.
- Ofrece un Framework de aplicaciones que se está habilitando para la reutilización y el reemplazo de componentes.
- La máquina virtual Dalvik, optimizada para dispositivos móviles.
- Navegador integrado, basado en el motor del proyecto abierto WebKit.
- Gráficos optimizados, suministrados por una biblioteca de gráficos 2D. Los gráficos 3D están basados en la especificación OpenGL ES 1.0, con soporte para aceleración gráfica por hardware (opcional).

- Cuenta con SQLite para estructurar el almacenamiento de datos.
- Soporte multimedia común para audio, video, imágenes, incluyendo varios formatos (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- Acceso a telefonía GSM, Bluetooth, EDGE, 3G, WiFi, Camara, GPS, compass y acelerómetro siempre y cuando el hardware lo soporte.
- Completo entorno de desarrollo que incluye un dispositivo emulador, herramientas de depuración, y un plugin para el IDE Eclipse.

3.2.1. Arquitectura de Android

En la figura N°12, se presenta la arquitectura de Android.

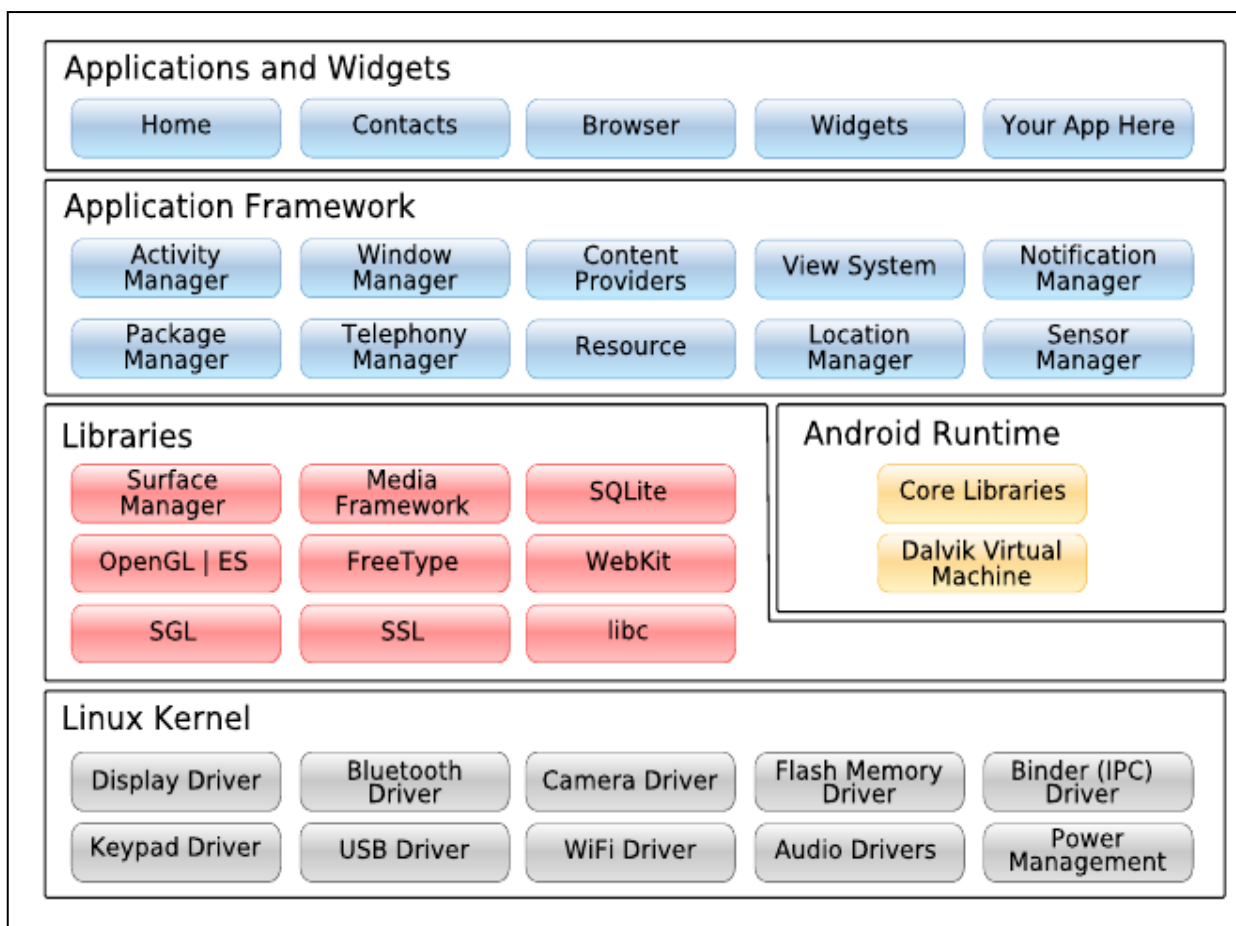


Figura 12, Arquitectura de Android (Introducción a la plataforma Android, 2010)

El detalle de la arquitectura se describe a continuación:

Aplicaciones: Las aplicaciones base incluyen un cliente de email, programa de SMS, calendario, mapas, navegador, contactos, y otros. Todas las aplicaciones están escritas en el lenguaje de programación Java.

Framework de aplicaciones: Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier usuario puede publicar sus aplicaciones para que otros hagan uso de estas (sujeto a reglas de seguridad del framework). Éste mismo mecanismo permite que los componentes sean reemplazados por el usuario. Una capa de servicios disponibles para las aplicaciones incluye:

- Un completo y extensible conjunto de vistas que pueden ser utilizadas para desarrollar una aplicación: listas, grillas, cajas de texto, botones e incluso un web browser.
- Proveedores de contenidos que permiten el acceso a datos provenientes de otras aplicaciones (como Contactos), o compartir sus propios datos.
- Un administrador de recursos, que provee acceso a cadenas, gráficos, y archivos.
- Un administrador de notificaciones que permite a toda aplicación mostrar alertas personalizables en la barra de estatus.
- Un administrador de actividades que maneja el ciclo de vida de las aplicaciones y provee un comportamiento común en la navegación.

Bibliotecas: Android incluye un conjunto de bibliotecas C/C++ usadas por varios componentes del sistema Android. Estas capacidades se exponen a los desarrolladores a través del framework de aplicaciones de Android. Algunas son: System C library (implementación bibliotecas C standard), bibliotecas de medios, bibliotecas de gráficos, 3d, SQLite, entre otras.

Runtime de Android: Android incluye un conjunto de bibliotecas base que proveen la mayor parte de las funciones disponibles en el lenguaje de programación Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de manera que un dispositivo puede correr en múltiples máquinas virtuales de forma eficiente. La Máquina Virtual está basada en registros, y corre

clases compiladas por el compilador de Java que han sido transformadas al formato .dex por la herramienta incluida "dx" [3].

Núcleo - Linux: Android depende de un Linux versión 2.6 para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, stack de red, y modelo de drivers. El núcleo también actúa como una capa de abstracción entre el hardware y el resto del stack de software.

3.3. FILOSOFÍA DE DESARROLLO ANDROID

3.3.1. Modelo Vista Controlador

Android sigue el patrón de diseño modelo vista controlador (MVC), la cual separa los datos de la aplicación, la interfaz de usuario y la lógica del negocio en tres componentes distintos.

Al crear una aplicación en Android automáticamente se crearán 3 carpetas la cuales permiten seguir el orden del modelo de diseño descrito anteriormente, las carpetas son las siguientes:

Carpeta src: Es el espacio en donde se crea toda la lógica del negocio. Aquí se ubican los archivos Java que implementa a las vistas de la aplicación (Activity).

Carpeta gen: Contiene un único archivo llamado R.java, el cual no debe ser modificado ya que es Eclipse quien se encarga de poner el código en él. Este archivo sirve básicamente para enlazar las tareas que se hacen en XML con la programación en Java.

Android Library: Es la referencia de Eclipse al sdk de Android.

Carpeta assets: Aquí se incluyen los archivos varios que puede ser música, pdf, zip, rar.

Carpeta res: Dentro de esta carpeta se encuentra el archivo AndroidManifest.xml. Todas las aplicaciones deben tener este archivo y no debe ser renombrado. En él se especifican las opciones generales del programa, como el paquete principal, la actividad que deberá ejecutarse al iniciar la aplicación (y deben incluirse allí TODAS las actividades que se van

usar), el icono a usar, los permisos, etc, además de este archivo hay tres subcarpetas las cuales guardan todo lo necesario para la interfaz, estas carpetas son:

- **Carpeta Drawable:** Contiene todos las figuras y dibujos que se utilizarán en la aplicación.
- **Carpeta layout:** Contiene las Activitys que presentan una interfaz gráfica, permite al usuario interactuar con la aplicación. Estas Activitys están escritas en XML.
- **Carpeta Value:** Está contenido inicialmente el archivo strings.xml, en el que se declara las cadenas de texto que usará la aplicación.

3.3.2. Lenguaje XML

XML (Extensible Markup Language), es un lenguaje diseñado especialmente para crear documentos en Web.

Sus objetivos son:

- XML debe ser directamente utilizable sobre Internet.
- XML debe soportar una amplia variedad de aplicaciones.
- XML debe ser compatible con SGML.
- Debe ser fácil la escritura de programas que procesen documentos XML.
- El número de características opcionales en XML debe ser absolutamente mínima, idealmente cero.
- Los documentos XML deben ser legibles por humanos y razonablemente claros.
- El diseño de XML debe ser preparado rápidamente.
- Los documentos XML deben ser fácilmente creables.
- La concisión en las marcas XML es de mínima importancia.

Básicamente Android utiliza este lenguaje para crear interfaces graficas que interactúen con el usuario (layout) y para ficheros de traducción.

3.3.3. Lenguaje Java

Java es un lenguaje orientado a objetos y es el utilizado para describir las aplicaciones en Android.

Java es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Es un lenguaje independiente de la plataforma en la que se trabaje, esta independencia es una de las razones por las que Java es interesante para Internet y el desarrollo de aplicaciones para dispositivos móviles.

Algunas ventajas de JAVA:

- Es una fuente abierta, así que los usuarios no tienen que incurrir en costos.
- Independiente de la plataforma.
- Java realiza la colección de basura de las ayudas, así que la administración de memoria es automática.
- Es simple y robusto.
- Usando JAVA se puede desarrollar aplicaciones Web dinámicas.
- Permite crear programas modulares y códigos reutilizables.

Android utiliza estas ventajas para el desarrollo ya que toda la parte lógica de las aplicaciones están descritas en este lenguaje, utilizando bibliotecas tanto propias como las proporcionadas por Google.

La única diferencia que existe entre el Java utilizado comúnmente y este destinado a Android, es que esta última utiliza una máquina virtual llamada Dalvik la cual está personalizada y ha sido diseñada para optimizar la memoria y los recursos de hardware en un entorno móvil.

Capítulo IV

DESARROLLO DE APLICACIÓN NATIVA SOBRE ANDROID

En este capítulo se abordarán temas pertinentes al desarrollo de la aplicación nativa sobre el sistema operativo Android. Se realizará basándose en las etapas de la ingeniería de software las cuales se componen de:

- Análisis de requisitos
- Diseño de la interfaz
- Modelado de datos (Entidad Relación)
- Diagramas de colaboración
- Patrones de diseño
- Mapas de navegación.

Cada uno de los puntos descritos anteriormente son realizados previo a la implementación del software con la finalidad de adquirir la información necesaria para construir un software de calidad.

4.1. ANÁLISIS DE REQUISITOS

El análisis de requisitos corresponde a la primera fase de desarrollo de un software, permite conocer mediante ciertas tareas las condiciones o funcionalidades que debe cumplir el software a construir.

Siempre se debe tener en cuenta la visión del usuario, pues será este quien finalmente haga uso de la aplicación (Somerville, 2005).

4.1.1. Requisitos no funcionales

Los requisitos no funcionales son aquellos relacionados con características que pueden afectar diversos aspectos de una aplicación, como por ejemplo, el rendimiento, mantenimiento, seguridad, portabilidad y estándares.

En la tabla N° 1 se describen los requisitos no funcionales para la aplicación a construir.

Atributo	Detalle
Plataforma	Android versión 1.5
Tipo de aplicación	Aplicación Móvil
Lenguaje de programación	Java y XML
Gestor de base de datos	SQLite
Herramienta de desarrollo	Eclipse
Tiempo de respuesta	No mayor a 7 segundos para cualquier operación.
Disponibilidad	En todo momento siempre y cuando se cuente con WIFI.
Metáfora de interfaz	Orientada a formulario y mapas

Tabla 1, Requisitos no funcionales

4.1.2. Requisitos funcionales

Los requisitos funcionales son la declaración de los servicios que debe proporcionar el sistema, cuya ejecución depende principalmente de módulos construidos en base a código y que han debido ser obligatoriamente incorporados. Aquellos servicios que el usuario conoce y puede percibir durante el uso de una aplicación reciben el nombre de **evidentes**, los que a su vez pueden acarrear una serie de servicios **ocultos** que la aplicación debe realizar de manera interna sin que el usuario tenga conocimiento de ello. Los servicios ocultos se destacan por no ser visibles y a menudo se omiten (erróneamente) durante el proceso de obtención de requerimientos (Larman, 1999).

Tomando en cuenta la definición de los requisitos funcionales, es inevitable omitir en su captura la participación del usuario. Debido a esto el equipo de desarrollo a cargo del proyecto ha optado por mantener desde un principio una constante comunicación con el personal médico del consultorio, específicamente con aquellos encargados de realizar atenciones en terreno.

Por esta razón se estableció contacto vía correo electrónico con la directora del Centro de Salud Familiar (CESFAM) señora Viviana Bustamante Rubilar, quien actualmente administra el consultorio Dr. Federico Puga Borne en la comuna de Chillán Viejo. Aquí, se optó por presentar al equipo de desarrollo, dar a conocer en forma resumida la naturaleza del proyecto y solicitar colaboración en aspectos técnicos que se desconozcan. La señora Viviana Bustamante, nos presentó a los miembros del personal encargado de la atención a domicilio, los señores Manuel Valdés y Pablo Caamaño ambos médicos generales junto a la señora Sandra Guajardo, secretaria y enfermera asesora del programa postrados quien además administra las horas médicas. Se procedió a informar a cada uno de los miembros presentes los objetivos que pretendía alcanzar el proyecto, ante lo cual se mostraron claramente motivados en participar y contribuir con sus experiencias.

Capítulo IV, Desarrollo aplicación nativa sobre Android

A continuación, en la tabla N° 2 se detallan los requisitos funcionales extraídos de una primera encuesta enviada a los miembros del personal médico del consultorio, dicha encuesta se encuentra detallada en el Anexo D, algunos requisitos son aseveraciones directas de los usuarios finales y otros se derivan de la naturaleza del proyecto.

Ref	Función	Categoría
R1	Validar usuario	Oculto
R2	Ver citas del día	Evidente
R3	Registrar Nueva ficha médica	Evidente
R4	Modificar Ficha Médica	Evidente
R5	Agregar información a ficha médica	Evidente
R6	Buscar citas por fecha	Oculto
R7	Buscar dirección	Oculto
R8	Buscar paciente	Oculto
R9	Ver actualizaciones	Evidente
R10	Ver información paciente	Evidente
R11	Marcar estado visitas	Evidente
R12	Cancelar visita médica	Evidente
R13	Agregar visita médica	Evidente
R14	Modificar visita médica	Evidente
R15	Envío de aviso de cancelación cita	Oculto
R16	Cerrar sesión	Evidente

Tabla 2, Requisitos Funcionales

4.1.3. Plantilla combinada

En la tabla N° 3, se muestran los atributos del sistema relacionados con los requisitos funcionales.

Ref.	Función	Categoría	Atributo	Detalle y restricciones	Categoría
R1	Validar usuario	Oculto	Tiempo de respuesta	3 seg.	Obligatorio
R2	Ver citas del día	Evidente	Tiempo de respuesta	5 seg.	Obligatorio
			Metáfora de Interfaz	Basado en colores.	
R3	Registrar Nueva ficha médica	Evidente	Tiempo de respuesta	5 seg.	Obligatorio
			Tolerancia a fallos	Entregar aviso cuando los datos no se guarden	
			Metáfora de interfaz	Basado en formularios.	
R4	Modificar ficha médica	Evidente	Tiempo de respuesta	3 seg.	Obligatorio
			Tolerancia a fallos	Entregar aviso cuando los datos no se guarden	
			Metáfora de interfaz	Basada en formularios	
R5	Agregar información a ficha médica	Evidente	Tiempo de respuesta	5 seg.	Obligatorio
			Tolerancia a fallos	Entregar aviso cuando los datos no se guarden	
			Metáfora de interfaz	Basada en formularios	
R6	Buscar citas por fecha	Oculto	Tiempo de respuesta	3 seg.	Obligatorio

Capítulo IV, Desarrollo aplicación nativa sobre Android

R7	Buscar dirección	Oculto	Tiempo de respuesta	10 seg.	Obligatorio
			Metáfora de interfaz	Basadas en mapas.	
R8	Buscar paciente	Oculto	Tiempo de respuesta	5 seg.	Obligatorio
R9	Ver actualizaciones	Evidente	Tiempo de respuesta	3 seg.	Obligatorio
R10	Ver información del paciente	Evidente	Tiempo de respuesta	3 seg.	Obligatorio
R11	Marcar estado de visitas	Evidente	Tiempo de respuesta	1 seg.	Obligatorio
R12	Cancelar visita médica	Evidente	Tiempo de respuesta	2 seg.	Obligatorio
R13	Agregar visita médica	Evidente	Tiempo de respuesta	3 seg.	Obligatorio
			Tolerancia a fallos	Entregar aviso cuando los datos no se guarden	
			Metáfora de interfaz	Basada en formularios	
R14	Modifica visita médica	Evidente	Tiempo de respuesta	3 seg.	Obligatorio
			Tolerancia a fallos	Entregar aviso cuando los datos no se guarden	
			Metáfora de interfaz	Basada en formularios	
R15	Envío de aviso de cancelación cita	Oculto	Tiempo de respuesta	5 seg.	Obligatorio
R16	Cerrar sesión	Evidente	Tiempo de respuesta	3 seg.	Obligatorio

Tabla 3, Plantilla combinada.

4.1.4. Identificación de actores del sistema

El único actor identificado para el sistema de Agenda Médica es el que se describe en la tabla N° 4:

Actor:	Médico en terreno
Descripción:	Profesional encargado de realizar estudio, diagnóstico y tratamiento de enfermedades o lesiones del paciente en su domicilio.
Características:	Podrá tener uso ilimitado de la aplicación contenida en el dispositivo móvil. Accediendo a todos los servicios ofrecidos en él, tales como ver fichas de pacientes, ver citas agendadas, registrar y modificar fichas, buscar pacientes y buscar direcciones mediante GPS.

Tabla 4, Actores del sistema

4.1.5. Modelo de casos de Uso

En la figura Nº 13 se representan los casos de usos identificados para el actor "Médico en terreno" de la agenda médica.



Figura 13, Diagrama de casos de uso.

4.1.5. Especificación de casos de uso

En cada una de las especificaciones de casos de uso que a continuación se detallan se utiliza el término usuario que no es más que una referencia al concepto del médico en terreno, esto por motivos de promover la simplicidad al lector.

Se asume que al momento de realizar cada una de las acciones especificadas en los casos de uso, se cuenta con conexión WIFI constante de lo contrario cada actividad estará sujeta a errores de conexión.

La tabla N° 5 que se describe a continuación detalla el caso de uso "Autenticar Usuario".

<u>Caso de Uso</u>	Autenticar Usuario.
<u>ID</u>	1
<u>Descripción</u>	El usuario ingresa sus datos de acceso para acceder al sistema.
<u>Actores del sistema</u>	Médico en terreno.
<u>Actores secundarios</u>	No tiene.
<u>Precondiciones</u>	Ninguna.
<u>Flujos principales</u>	<ol style="list-style-type: none"> 1.-El usuario selecciona un tipo de ingreso 2.- El usuario ingresa el(los) dato(s) requerido(s). 3.-El sistema verifica que los datos sean correctos. <ol style="list-style-type: none"> 3.a.-Si los datos son correctos, el sistema despliega la actividad principal con las visitas diarias. 3.b.-Si los datos ingresados no son correctos, el sistema despliega un mensaje de error el cual debe ser confirmado por el usuario y así retornar al punto 1.
<u>Post-condiciones</u>	Ingreso a la aplicación.

Tabla 5, Autenticar usuario

Capítulo IV, Desarrollo aplicación nativa sobre Android

La tabla N° 6 detalla el caso de uso "Ver visitas del día".

<u>Caso de Uso</u>	Ver visitas del día.
<u>ID</u>	2
<u>Descripción</u>	El usuario revisa el detalle de las citas asignadas.
<u>Actores del sistema</u>	Médico en terreno.
<u>Actores secundarios</u>	No tiene.
<u>Precondiciones</u>	Tener citas asignadas.
<u>Flujos principales</u>	<p>1.-El sistema despliega las citas agendadas para el día.</p> <p>1.a.-En caso de no existir visitas para el día, el sistema avisa al usuario mediante un mensaje.</p> <p>2.- El usuario despliega más detalle de los pacientes a visitar, pulsando sobre ellas.</p> <p>3.- El sistema despliega la información detallada del paciente pulsado. (Véase caso de uso "Ver perfil paciente")</p>
<u>Post-condiciones</u>	Información de todas las visitas registradas para el día.

Tabla 6, Ver visitas del día

La tabla N° 7 detalla el caso de uso "Ver ficha médica".

<u>Caso de Uso</u>	Ver ficha médica
<u>ID</u>	3
<u>Descripción</u>	El usuario ve los datos de una ficha médica registrada.
<u>Actores del sistema</u>	Médico en terreno
<u>Actores secundarios</u>	No tiene.
<u>Precondiciones</u>	<p>1.- El usuario deberá estar autenticado en el sistema (véase caso de uso Autenticar Usuario).</p> <p>2.- La ficha médica a revisar deberá estar registrada previamente en el sistema (véase caso de uso Crear ficha médica).</p>
<u>Flujos principales</u>	<p>1.- El usuario selecciona la opción ver ficha médica.</p> <p>2.- El sistema despliega el detalle de la ficha médica con todos los datos registrados del paciente.</p>
<u>Post- condiciones</u>	Detalle de la ficha médica del paciente

Tabla 7, Ver ficha médica

Capítulo IV, Desarrollo aplicación nativa sobre Android

La tabla N° 8 describe el detalle del caso de uso "Buscar visita por fecha".

<u>Nombre</u>	Buscar citas por fecha
<u>ID</u>	4
<u>Descripción</u>	El usuario busca una cita especificando una fecha.
<u>Actores Principales</u>	Médico en terreno
<u>Actores Secundarios</u>	No tiene.
<u>Precondiciones</u>	1. El usuario deberá estar autenticado en el sistema (véase caso de uso Autenticar Usuario).
<u>Flujo Principal</u>	1. El caso de uso inicia cuando el usuario desea buscar las visitas en una fecha específica. 2.- El usuario ingresa la fecha de las visitas a buscar. 3.- El sistema busca la visitas para el día seleccionado. (véase caso de uso "Ver visitas del día")
<u>Post- condición</u>	Detalle de todas las visitas del día escogido.

Tabla 8, Buscar citas por fecha.

La tabla N° 9 detalla el caso de uso "Ver historial de observaciones"

<u>Caso de Uso</u>	Ver historial de observaciones
<u>ID</u>	5
<u>Descripción</u>	El usuario ve el historial de observaciones realizadas al paciente
<u>Actores del sistema</u>	Médico en terreno
<u>Actores secundarios</u>	No tiene.
<u>Precondiciones</u>	1.- El usuario deberá estar autenticado en el sistema (véase caso de uso Autenticar Usuario). 2.- El paciente debe tener observaciones registradas.
<u>Flujos principales</u>	1.- El usuario selecciona ver historial. 2.- El sistema despliega el historial de visitas. 2.2.-En caso de no haber visitas para el mes indicado el sistema avisa mediante un mensaje.
<u>Post- condiciones</u>	Lista del historial de observaciones.

Tabla 9, Ver historial de observaciones

Capítulo IV, Desarrollo aplicación nativa sobre Android

La tabla N° 10, describe la especificación del caso de uso "Crear ficha médica".

<u>Nombre</u>	Crear ficha médica.
<u>ID</u>	6
<u>Descripción</u>	Crear el registro de una nueva ficha médica para paciente
<u>Actores Principales</u>	Médico en terreno
<u>Actores Secundarios</u>	No tiene.
<u>Precondiciones</u>	<p>1.- El usuario deberá estar autenticado en el sistema (véase caso de uso Autenticar Usuario).</p> <p>2.- El registro del paciente no existe en el sistema.</p> <p>3.- Se cuenta con todos los datos del paciente que el sistema requiere para su registro.</p>
<u>Flujos Principales</u>	<p>1.- El caso de uso inicia cuando el usuario desea registrar una nueva ficha médica para un paciente y ha seleccionado esta opción.</p> <p>2.- El sistema despliega por pantalla el formulario de ingreso para nuevas fichas médicas para pacientes.</p> <p>3.- El usuario ingresa en el formulario los datos del paciente priorizando los de carácter obligatorio.</p> <p>4.- El usuario selecciona la opción Guardar.</p> <p>5.- El sistema verifica el llenado de todos los campos obligatorios:</p> <p>5.a.- Si los campos obligatorios son llenados procede al paso 6.</p> <p>5.b.- Si falta algún campo obligatorio, el sistema avisa mediante un mensaje de error los campos faltantes. Posteriormente el usuario debe confirmar dicho mensaje y retornar al paso 3.</p> <p>6.- El sistema verifica que los datos ingresados sean correctos.</p> <p>6.a.- Si todos los datos son consistentes procede al paso 7.</p> <p>6.b.- Si algún dato está mal ingresado, el sistema avisa mediante un mensaje el cual debe ser confirmado para volver al paso 3.</p> <p>7.- El sistema confirma mediante un mensaje el éxito de la operación.</p>
<u>Post-condiciones</u>	La ficha médica del paciente queda guardada en el sistema.

Tabla 10, Crear ficha médica

Capítulo IV, Desarrollo aplicación nativa sobre Android

La tabla N° 11, el caso de uso "Agregar observación médica".

<u>Nombre</u>	Agregar observación médica.
<u>ID</u>	7
<u>Descripción</u>	Agrega observaciones de la visita realizada a una ficha médica existente.
<u>Actores Principales</u>	Médico en terreno
<u>Actores Secundarios</u>	No tiene.
<u>Precondiciones</u>	1. El usuario deberá estar autenticado en el sistema (véase caso de uso Autenticar Usuario). 2. La ficha médica debe existir en el sistema.
<u>Flujos Principales</u>	1.- El caso de uso inicia cuando el usuario ingresa observaciones de la visita a una ficha médica existente. 2.-El sistema despliega por pantalla un formulario de ingreso de datos. 3.-El usuario ingresa los datos pertinentes en el formulario. 4. - El usuario selecciona la opción Guardar. 5.- El sistema confirma mediante un mensaje el éxito de la operación.
<u>Post-condiciones</u>	Las observaciones quedan guardadas en la ficha médica del paciente.

Tabla 11, Agregar observación medica

La tabla N° 12, que se describe a continuación detalla el caso de uso "Cerrar sesión".

<u>Caso de Uso</u>	Cerrar sesión
<u>ID</u>	8
<u>Descripción</u>	El usuario cierra su sesión
<u>Actores del sistema</u>	Médico en terreno
<u>Actores secundarios</u>	No tiene.
<u>Precondiciones</u>	1.- El usuario deberá estar autenticado en el sistema (véase caso de uso Autenticar Usuario).
<u>Flujos principales</u>	1.- El usuario selecciona la opción cerrar sesión. 2.- El sistema consulta la decisión mediante un mensaje. 3.- El usuario confirma el mensaje. 4.- El sistema cierra la aplicación.
<u>Post- condiciones</u>	El usuario cierra su sesión.

Tabla 12, Cerrar sesión.

Capítulo IV, Desarrollo aplicación nativa sobre Android

La tabla N° 13, describe el caso de uso "Buscar dirección en el mapa"

<u>Nombre</u>	Buscar dirección en el mapa
<u>ID</u>	9
<u>Descripción</u>	El usuario busca una dirección en el mapa mediante el uso del GPS.
<u>Actores Principales</u>	Médico en terreno
<u>Actores Secundarios</u>	No tiene.
<u>Precondiciones</u>	1. El usuario deberá estar autenticado en el sistema (véase caso de uso Autenticar Usuario). 2.- Estar ubicado en el perfil del paciente.
<u>Flujo Principal</u>	1. El caso de uso inicia cuando el usuario desea ver la dirección del paciente en el mapa. 2.- El usuario selecciona "ver mapa". 3.- El sistema despliega un mapa con la ubicación de la dirección del paciente. 3.2.- En caso de no encontrar la dirección en el mapa, el sistema informa al usuario mediante un mensaje de error, volviendo al perfil del paciente.
<u>Post- condición</u>	Mapa con la ubicación visual de la dirección del paciente.

Tabla 13, Buscar Dirección

La tabla N° 14, describe el caso de uso "Llamar a un paciente"

<u>Nombre</u>	Llamar a un paciente
<u>ID</u>	10
<u>Descripción</u>	Llamar a un paciente registrado en el sistema
<u>Actores Principales</u>	Médico en terreno
<u>Actores Secundarios</u>	No tiene.
<u>Precondiciones</u>	1. El usuario deberá estar autenticado en el sistema (véase caso de uso Autenticar Usuario).
<u>Flujo Principal</u>	1. El caso de uso inicia cuando el usuario desea llamar a un paciente ingresado en el sistema. 2.- El médico selecciona la opción llamar. 3.- El sistema marca automáticamente el número telefónico del paciente.
<u>Post- condición</u>	Se realiza la llamada al paciente

Tabla 14, Llamar un paciente

Capítulo IV, Desarrollo aplicación nativa sobre Android

La tabla N° 15, detalla el caso de uso "Marcar estado de visitas"

<u>Nombre</u>	Marcar estado de visitas
<u>ID</u>	11
<u>Descripción</u>	El usuario marca estados de visitas para saber si fueron o no visitadas.
<u>Actores Principales</u>	Médico en terreno
<u>Actores Secundarios</u>	No tiene.
<u>Precondiciones</u>	1. El usuario deberá estar autenticado en el sistema (véase caso de uso Autenticar Usuario). 2.- Existir visitas médicas.
<u>Flujo Principal</u>	1. El caso de uso inicia cuando el usuario desea cambiar el estado de una visita médica 2.- El usuario pulsa el botón de estado de cada visita médica y la cambia (Visitada, pendiente o cancelada). 3.- El sistema cambia el estado dependiendo de la elección realizada por el usuario.
<u>Post- condición</u>	Estado de la visita cambiada.

Tabla 15, Marcar estados de visitas

Capítulo IV, Desarrollo aplicación nativa sobre Android

La tabla N° 16 que se describe a continuación detalla el caso de uso “Buscar paciente”

<u>Nombre</u>	Buscar paciente
<u>ID</u>	12
<u>Descripción</u>	El usuario busca un paciente por un determinado parámetro.
<u>Actores Principales</u>	Médico en terreno
<u>Actores Secundarios</u>	No tiene.
<u>Precondiciones</u>	<ol style="list-style-type: none"> 1. El usuario deberá estar autenticado en el sistema (véase caso de uso Autenticar Usuario). 2. Se cuenta con alguno de los parámetros base para realizar búsqueda.
<u>Flujo Principal</u>	<ol style="list-style-type: none"> 1.- El caso de uso inicia cuando el usuario desea buscar un paciente en el sistema y ha seleccionado la opción Buscar paciente. 2.- El sistema despliega el área de especificación para búsquedas, la cual consta de controles para seleccionar el parámetro de búsqueda, campos de ingreso para el mismo y la opción para iniciar la búsqueda. 3.- El usuario selecciona uno de los parámetros de búsqueda. 4.- El usuario ingresa el parámetro de búsqueda. 5. - El usuario selecciona la opción buscar. 6.- El sistema realiza la búsqueda. <ol style="list-style-type: none"> 6.1.- Si existen resultados, los despliega por pantalla. 6.2.- Si no existen resultados procede al paso 7. 7.- El sistema informa al usuario de la inexistencia del o los pacientes buscados mediante un mensaje, el cual debe ser confirmado para luego retornar al paso 4.
<u>Post- condición</u>	<ol style="list-style-type: none"> 1. En el caso de tener éxito en la búsqueda, el sistema desplegará el listado de el o las coincidencias encontradas, pudiendo el usuario tener más detalle pulsando sobre ellas.

Tabla 16, Buscar paciente

Capítulo IV, Desarrollo aplicación nativa sobre Android

La tabla N° 17 que se describe a continuación detalla el caso de uso "Ver perfil del paciente"

<u>Nombre</u>	Ver perfil del paciente
<u>ID</u>	13
<u>Descripción</u>	El usuario revisa el perfil de un paciente.
<u>Actores Principales</u>	Médico en terreno
<u>Actores Secundarios</u>	No tiene.
<u>Precondiciones</u>	<ol style="list-style-type: none"> 1. El usuario deberá estar autenticado en el sistema (véase caso de uso Autenticar Usuario). 2.- El paciente deberá existir en el sistema. 3.- Haber realizado una búsqueda del paciente (véase caso de uso "buscar paciente") o tener el listado de citas de pacientes.
<u>Flujo Principal</u>	<ol style="list-style-type: none"> 1.-El caso de uso inicia cuando el usuario desea ver el perfil de un paciente. 2.- El usuario pulsa sobre el paciente ya sea el encontrado en la búsqueda o el obtenido del listado de citas. 3.- El sistema muestra la información del paciente con los datos principales de este.
<u>Post- condición</u>	Información personal del paciente.

Tabla 17, Ver perfil del paciente

Capítulo IV, Desarrollo aplicación nativa sobre Android

La tabla N° 18, detalla el caso de uso "Agregar visita médica".

<u>Nombre</u>	Agregar visita médica.
<u>ID</u>	14
<u>Descripción</u>	Agrega una nueva visita médica en alguna hora libre.
<u>Actores Principales</u>	Médico en terreno
<u>Actores Secundarios</u>	No tiene.
<u>Precondiciones</u>	1. El usuario deberá estar autenticado en el sistema (véase caso de uso Autenticar Usuario). 2. La hora a agendar debe encontrarse libre de visitas.
<u>Flujos Principales</u>	1.- El caso de uso inicia cuando el usuario desea ingresar una nueva visita médica. 2.-El sistema despliega por pantalla un formulario de ingreso de datos para la visita médica. 3.-El usuario ingresa los datos pertinentes y selecciona la opción Guardar. 4.- El sistema verifica el llenado de todos los campos obligatorios: 4.1.-Si los datos obligatorios son llenados procede al paso 5. 4.2.- Si falta algún campo obligatorio, el Sistema avisa mediante un mensaje de error los campos faltantes. Posteriormente el usuario debe confirmar dicho mensaje y retornar al paso 3. 5.-El sistema verifica que los datos ingresados sean correctos. 5.1.- Si todos los datos son consistentes procede al paso 6. 5.2.- Si algún dato está mal ingresado, el sistema avisa mediante un mensaje el cual debe ser confirmado para volver al paso 3. 6.- El sistema confirma con un mensaje de éxito la operación.
<u>Post-condiciones</u>	La nueva visita médica queda agendada.

Tabla 18, Agregar visita médica.

4.2 DISEÑO DE PROTOTIPOS DE USABILIDAD

Para el desarrollo de la interfaz de la Agenda Médica, y para cualquier aplicación es necesario tener en cuenta los conceptos de usabilidad los cuales permiten entregar al usuario un software intuitivo, fácil de aprender y utilizar.

Teniendo claro que la Agenda Médica es una aplicación que estará contenida en un dispositivo móvil, se debe tener aún más cuidado, ya que estos aparatos poseen características que pueden jugar en contra de la usabilidad, como el limitado tamaño de la pantalla lo que obliga a mostrar información debidamente seleccionada, pocos mecanismos de entrada, capacidades reducidas y sistemas operativos (Portillo. J y Carretero N., 2010).

Se ha optado por un diseño sencillo y minimalista, haciendo el máximo uso de componentes que ayuden a la simplicidad de uso y el fácil entendimiento.

Algunos aspectos importantes que se deben considerar para una interfaz de una aplicación móvil son las que se describen a continuación.

- Mantener los iconos pequeños y reconocibles, para que el usuario pueda saber sin dificultad qué representa cada uno.
- Usar colores adecuados, para evitar problemas de visibilidad en condiciones de poca luz o entornos soleados.
- Los iconos deben ser simples, evitando combinaciones de dibujos.
- Los nombres son más fácilmente identificables que los verbos.
- Los iconos deben ser lo más consistentes posible, es decir, los iconos que ya sean reconocidos por su asociación a ciertas funciones no deben ser cambiados.
- No se debe usar texto dentro de los iconos, pues debido al tamaño su lectura no será adecuada.
- Uso de menús que faciliten el acceso al usuario a funcionalidades del dispositivo.
- Hacer uso de Check Box o componente similar, cuando sea posible, para evitar el ingreso manual de datos.
- En lo posible, se debe proveer de una opción de "Cancelar" o "Parar".

Capítulo IV, Desarrollo aplicación nativa sobre Android

- Uso de barras de progreso, para indicar al usuario esperas.
- Informar al usuario operaciones exitosas o fallidas.

Se desarrollaron prototipos de pantalla aplicando las técnicas de usabilidad anteriormente descritas. Dichas pantallas son las que se muestran en lo que sigue de esta sección.

La figura N° 14, muestra los dos tipos de entrada a utilizar en el teléfono, uno mediante lectura de un código QR y otro por ingreso de parámetros. Esta última es más lenta, pero ofrece una alternativa al uso del código QR y es más tradicional.

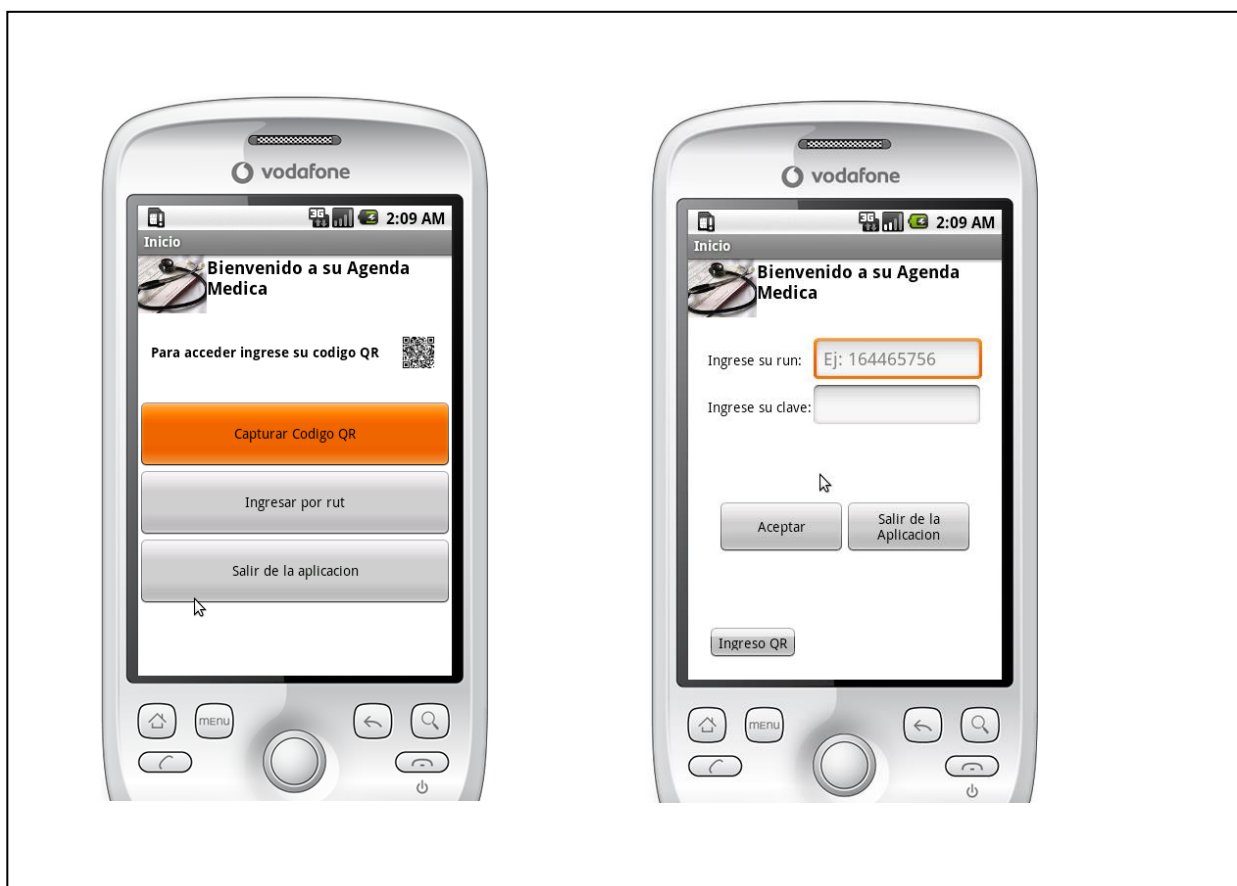


Figura 14, Interfaz de ingreso a la aplicación

Capítulo IV, Desarrollo aplicación nativa sobre Android

En la figura N° 15, se puede apreciar claramente el uso de componentes que brindan una mayor facilidad de uso al usuario, como por ejemplo el selector de fechas, la utilización de menús de atajos, campos de entrada de datos predefinidos y el uso de información seleccionada para evitar la sobrecarga de la vista que puede crear confusión en el usuario.



Figura 15, Componentes de usabilidad

Capítulo IV, Desarrollo aplicación nativa sobre Android

La figura Nº 16, contiene dos ejemplos, el primero, cómo se realizan las presentaciones de datos en la agenda médica y la segunda (figura del lado derecho) representa una forma de efectuar acciones haciendo uso de los distintos diálogos que ofrece Android.



Figura 16, Presentación de datos

Capítulo IV, Desarrollo aplicación nativa sobre Android

La figura N° 17, muestra el prototipo de ingreso de datos para la creación de fichas médicas. Según los conceptos de usabilidad, este diseño no es el más adecuado, puesto que el uso de muchos campos de entrada dificulta la tarea del usuario, obligando a este a invertir más tiempo para el realizar una tarea.

Desafortunadamente, el uso de este tipo de ingresos para crear una ficha médica es inevitable, pues se requiere contar con datos reales, entregados por el paciente al médico.



Figura 17, Interfaz de ficha médica

Capítulo IV, Desarrollo aplicación nativa sobre Android

La figura N° 18, contiene el prototipo utilizado para la representación gráfica de la dirección de un paciente en el mapa, haciendo uso de las bibliotecas ofrecidas por Google Maps. Se ha establecido que la ubicación actual del médico se represente por un punto azul y la dirección del paciente por una flecha negra, ambos símbolos unidos por una línea verde que indica la ruta a seguir para llegar a destino (dirección del paciente).

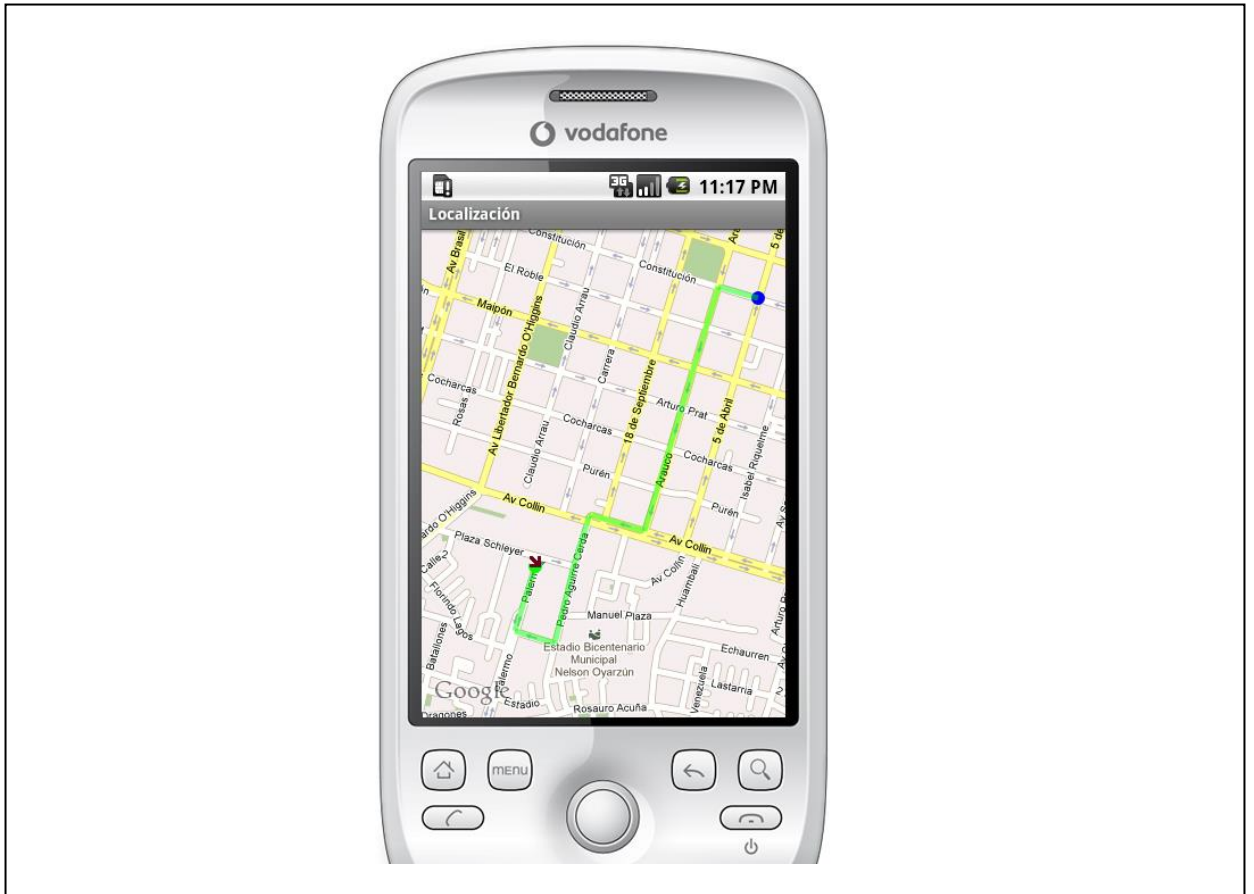


Figura 18, Representación de dirección en el Mapa

4.3. MODELO ENTIDAD RELACIÓN

El modelo de entidad relación, es una herramienta usada para el modelado de datos de una aplicación, con él se consigue representar de manera gráfica la estructura lógica de una base de datos.

En la figura N° 19 se representa el modelo entidad relación de la agenda médica.

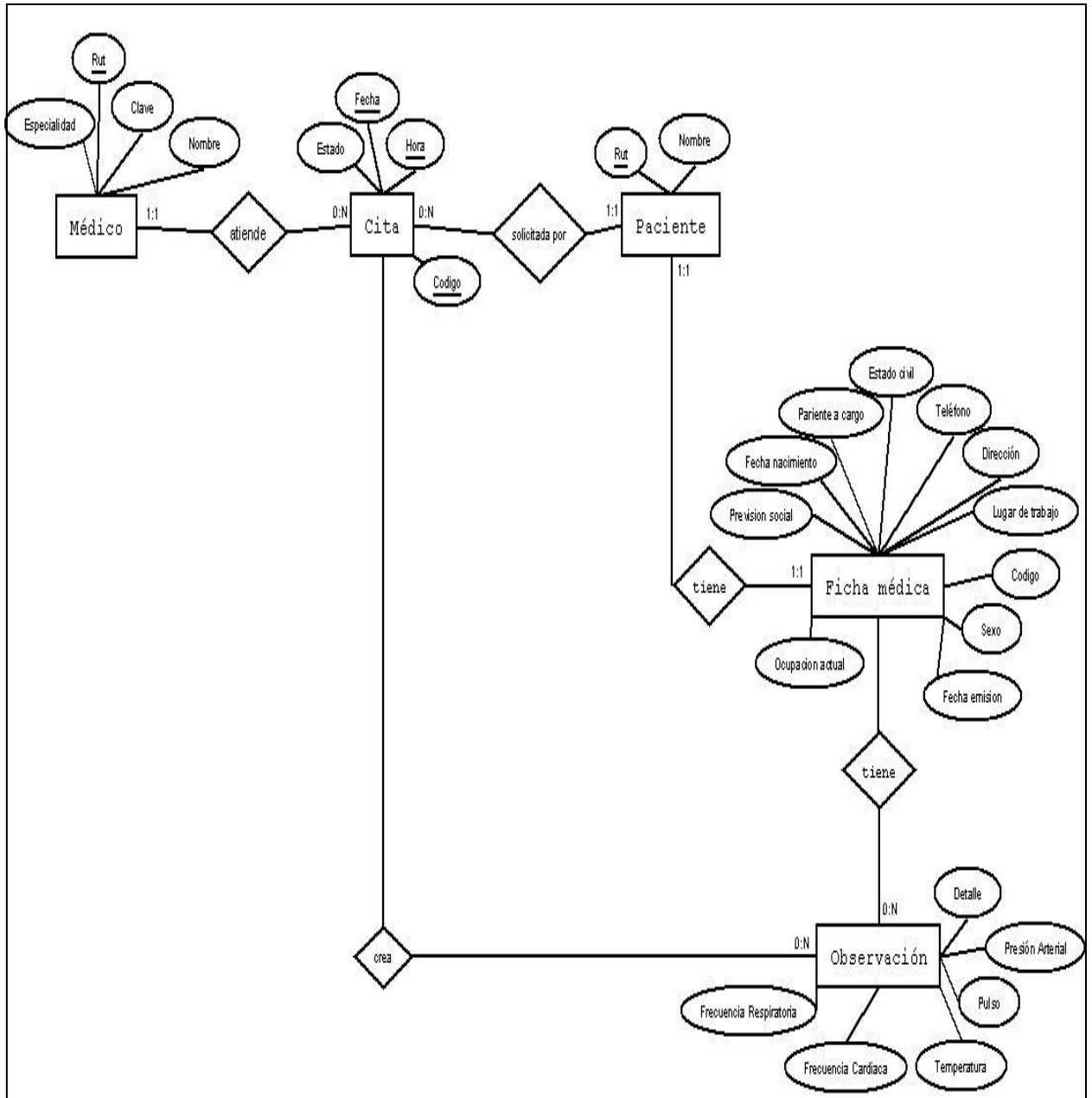


Figura 19, Modelo Entidad Relación

4.4. MODELO CONCEPTUAL

La figura N° 20 presenta el modelo conceptual del proyecto, en donde se aprecian las relaciones entre los distintos conceptos que intervienen en la aplicación.

El médico es el encargado de administrar y crear las fichas médicas pertenecientes a los pacientes y son estos los que solicitan visitas domiciliarias, las cuales son atendidas por algún médico tratante de turno.

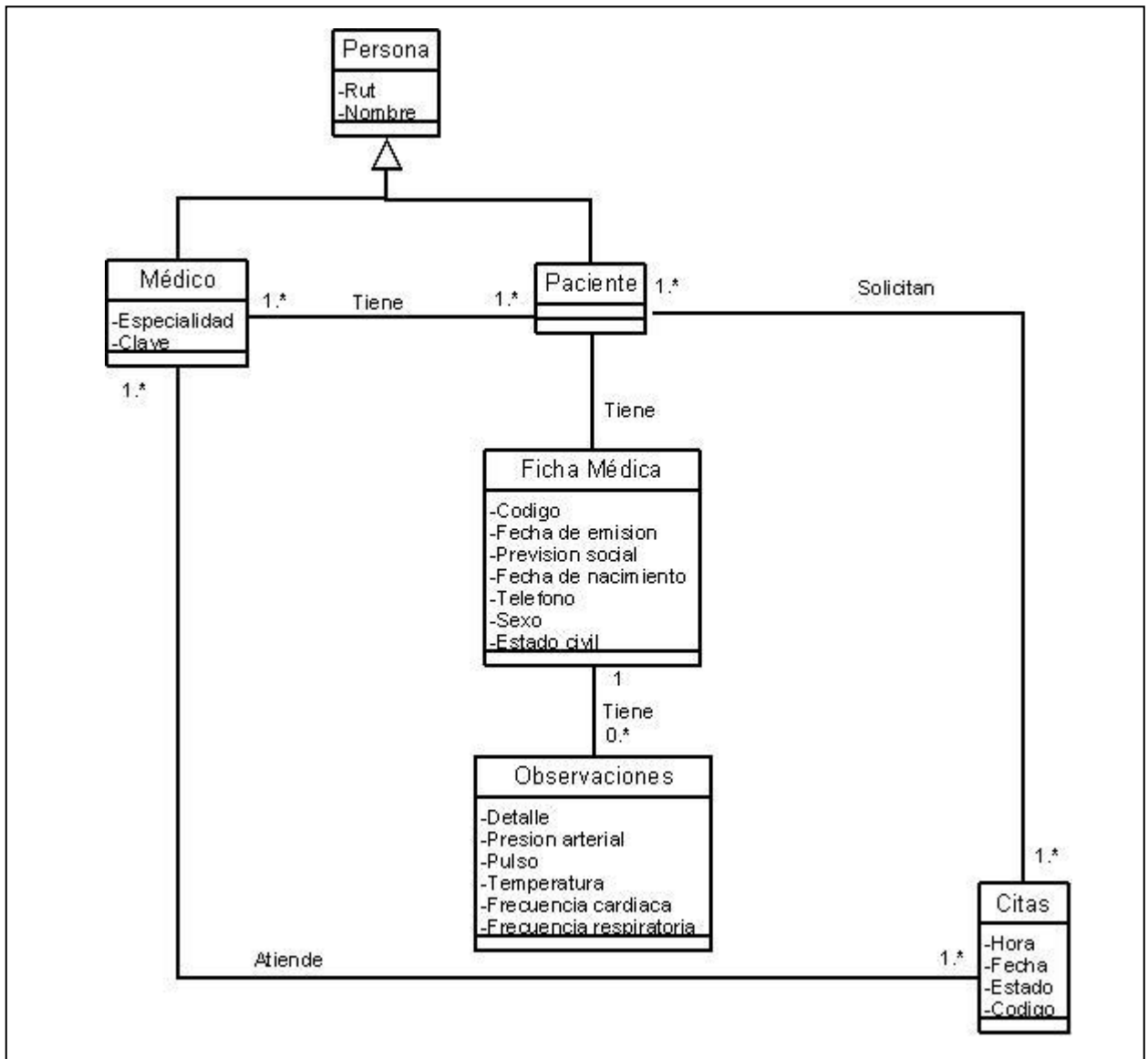


Figura 20, Modelo Conceptual

4.5. DIAGRAMA DE DESPLIEGUE

El proyecto desarrollado consta de dos subsistemas fundamentales los que colaboran entre sí para el correcto funcionamiento de la aplicación. Para que el lector entienda lo antes expuesto en la figura N° 21 se representa un diagrama de despliegue el cual muestra en la distribución de los procesos y componentes los nodos procesadores (Larman,1999, p.430).

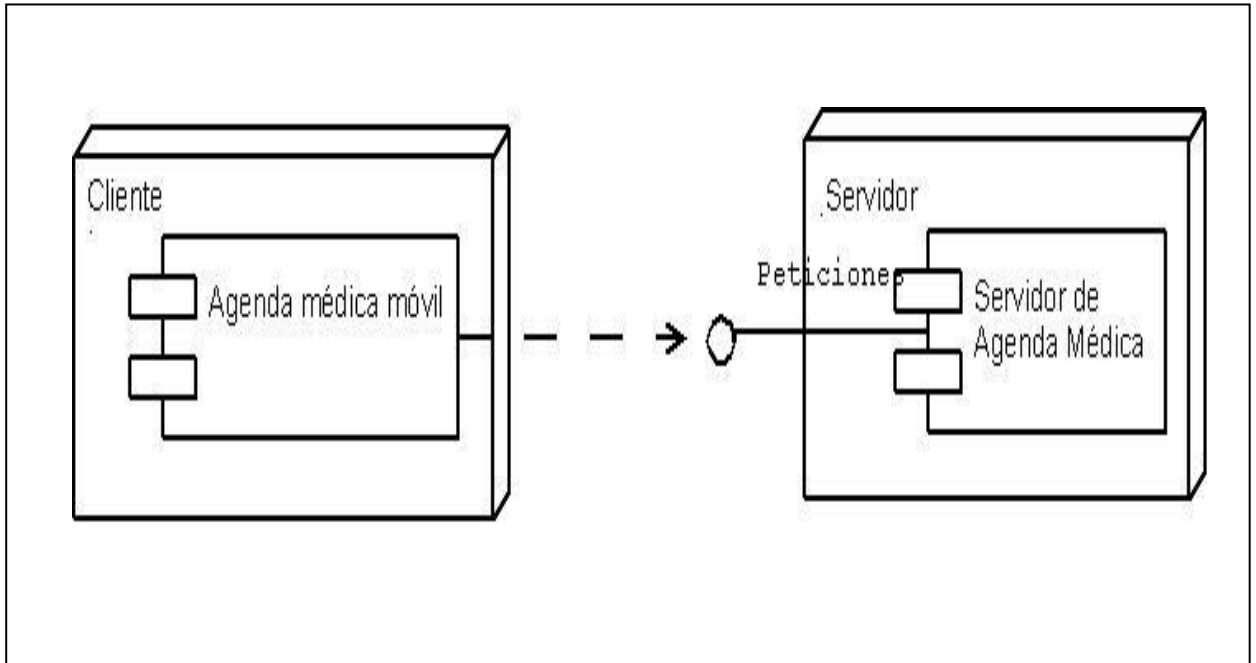


Figura 21, Diagrama de despliegue

La agenda médica móvil es el subsistema contenido en el SmartPhone, implementa la interfaz gráfica con la que interactúa el médico y es el encargado de enviar peticiones al servidor, este último se aloja en un computador y responde a cada una de las solicitudes realizadas desde el teléfono.

Capítulo IV, Desarrollo aplicación nativa sobre Android

4.7. DIAGRAMAS DE CLASE DEL DISEÑO

La figura N° 22 representa el diagrama de clase del diseño para cada uno de los subsistemas anteriormente descritos.

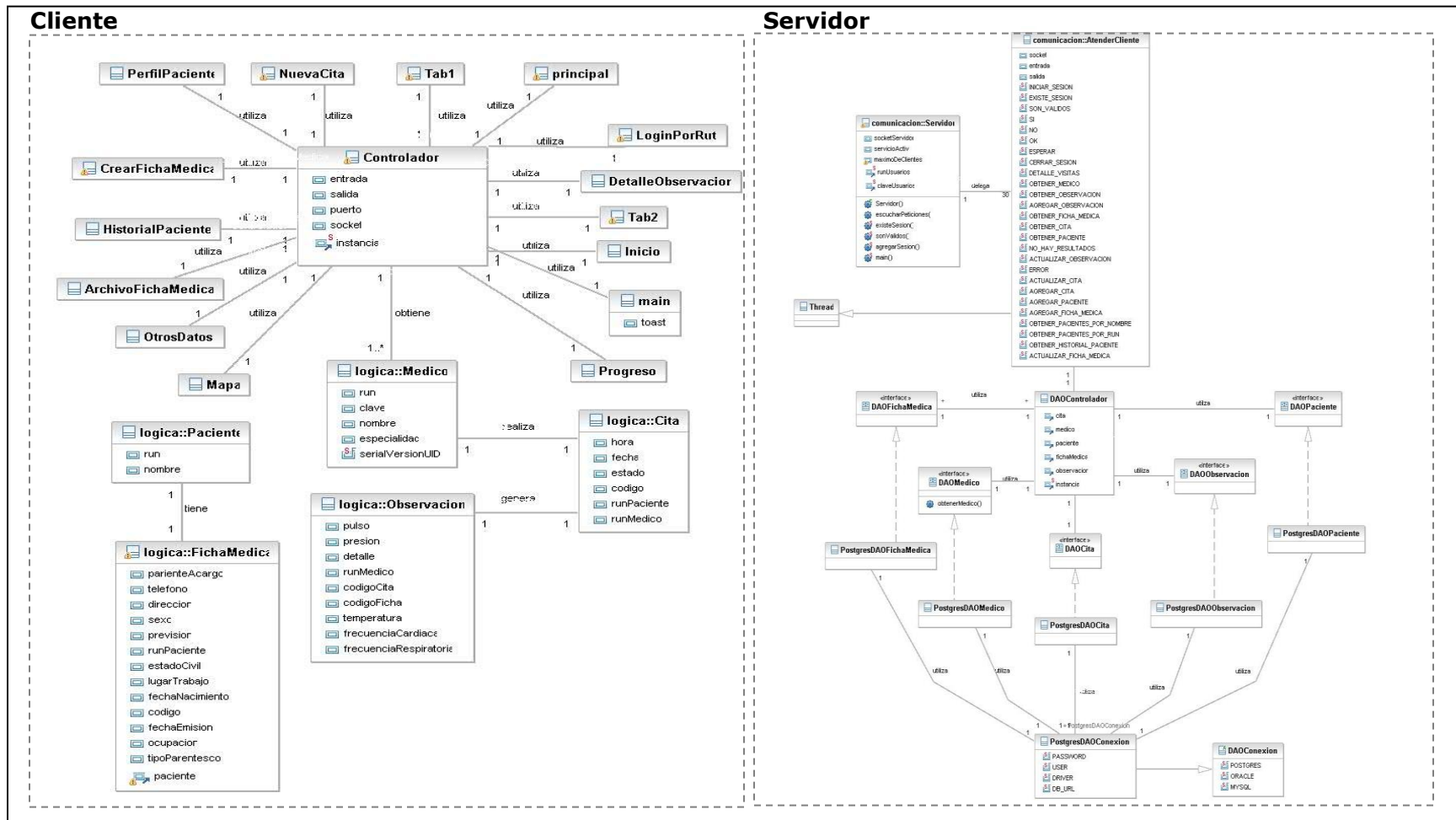


Figura 22, Diagrama de clase del diseño

4.7. DIAGRAMAS DE COLABORACIÓN

Los diagramas de colaboración muestran las interacciones que ocurren entre los objetos que participan en una situación determinada, fijando el interés en las relaciones entre los objetos y su topología (Pressman, 1999).

A continuación se detallan algunos diagramas de colaboración asociados a las interacciones que se producen en el servidor.

La figura N° 23 corresponde al diagrama *escucharPeticones*, en éste el servidor espera por alguna solicitud proveniente del cliente (teléfono), el resultado de la conexión entre ambas partes es un socket, por medio del cual es atender cada una de las peticiones realizadas.

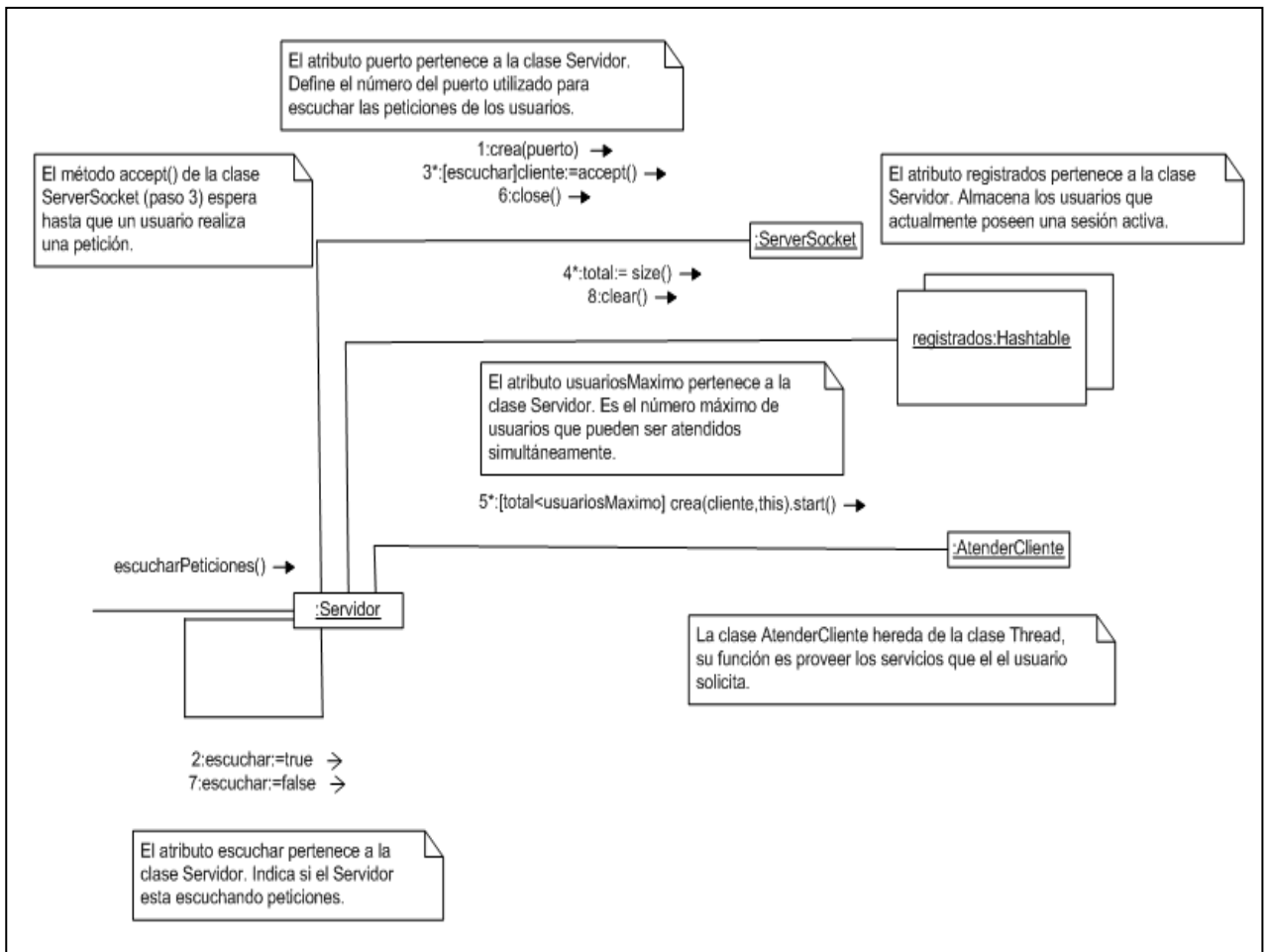


Figura 23, Diagrama Escuchar peticiones

Capítulo IV, Desarrollo aplicación nativa sobre Android

El diagrama de colaboración de la figura N° 24 corresponde a las interacciones que se producen al crear un objeto de la clase *atenderCliente*.

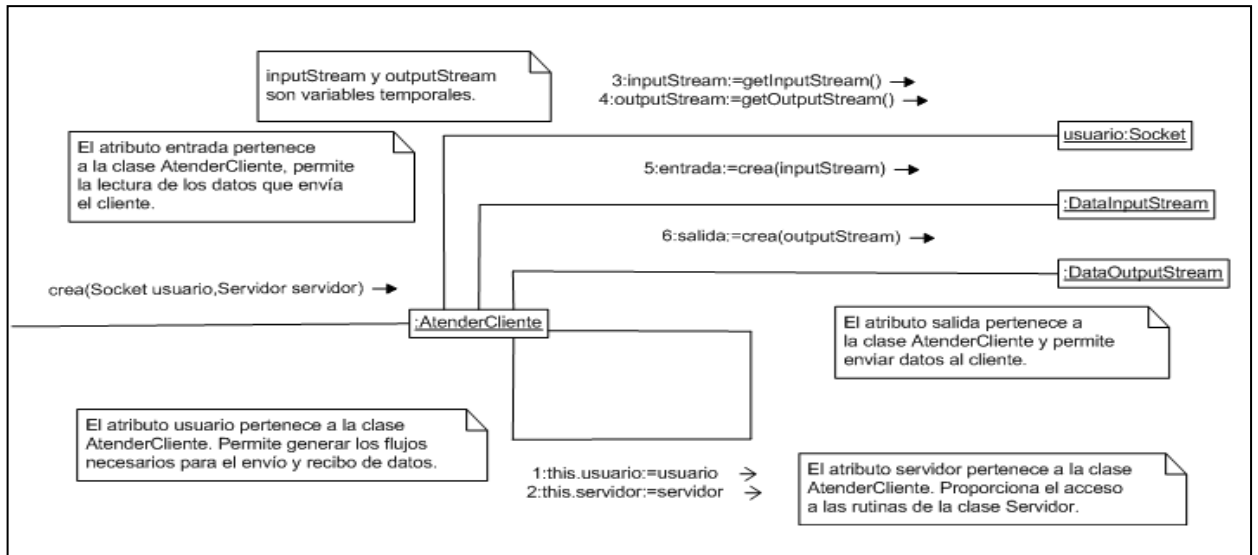


Figura 24, Diagrama Atender Cliente

Los diagramas que a continuación se detallan corresponden a las interacciones que se generan en la aplicación alojada en el teléfono.

La figura N° 25, detalla el diagrama de colaboración *abrirConexion*, el cual establece una conexión entre el teléfono y el servidor.

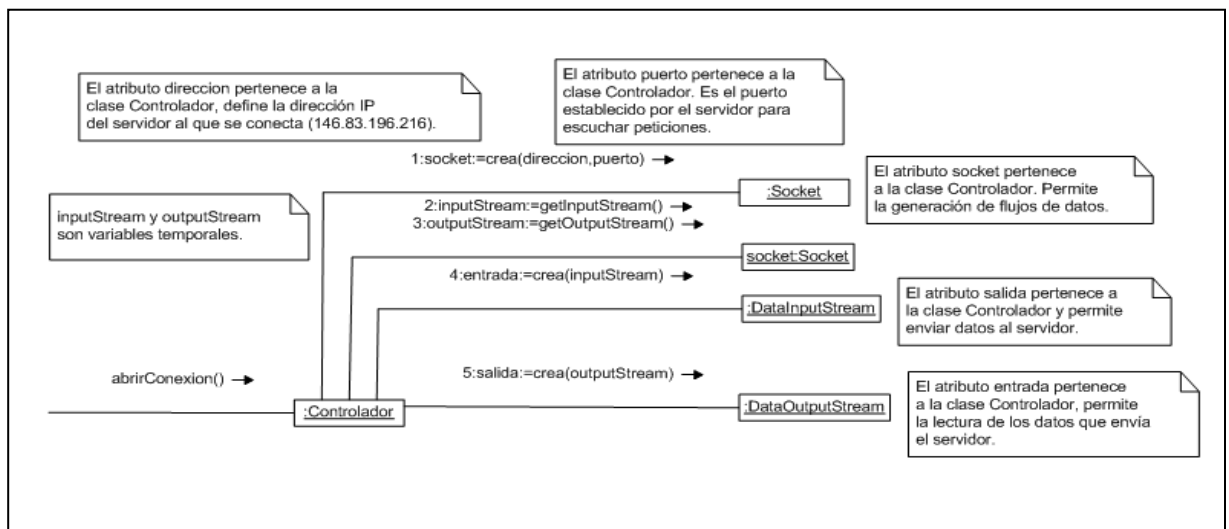


Figura 25, Diagrama abrirConexion

Capítulo IV, Desarrollo aplicación nativa sobre Android

La figura N° 26, detalla el diagrama de colaboración *estaConectado*, que permite a la aplicación alojada en el teléfono verificar si existe una conexión con el servidor.

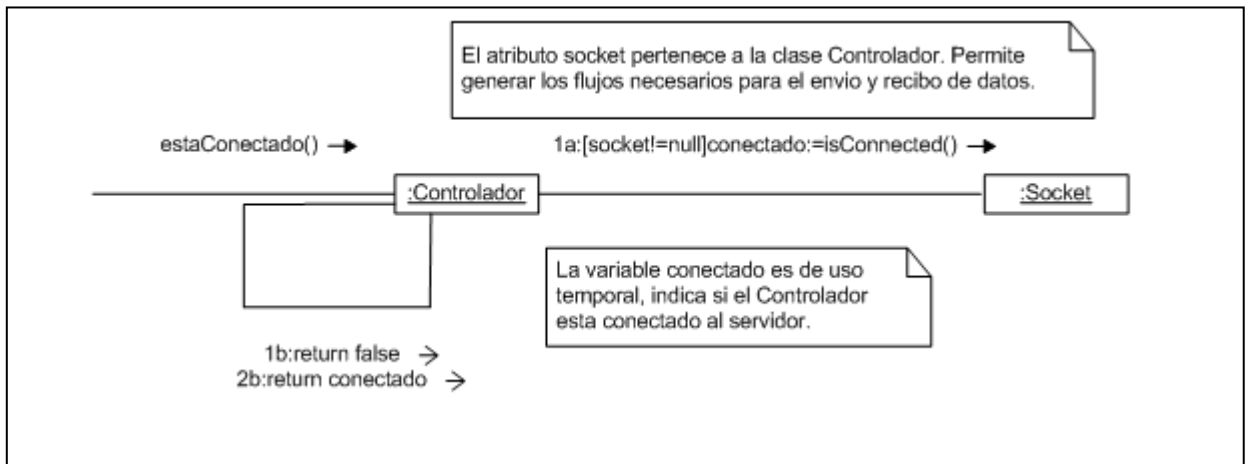


Figura 26, Diagrama estaConectado

La figura N° 27, detalla el diagrama de colaboración *sonValidos*, que chequea la validez de un determinado run y clave pasados por parámetro.

Este tipo de procedimiento se lleva a cabo con la mayoría de las consultas que se le realizan desde el cliente al servidor.

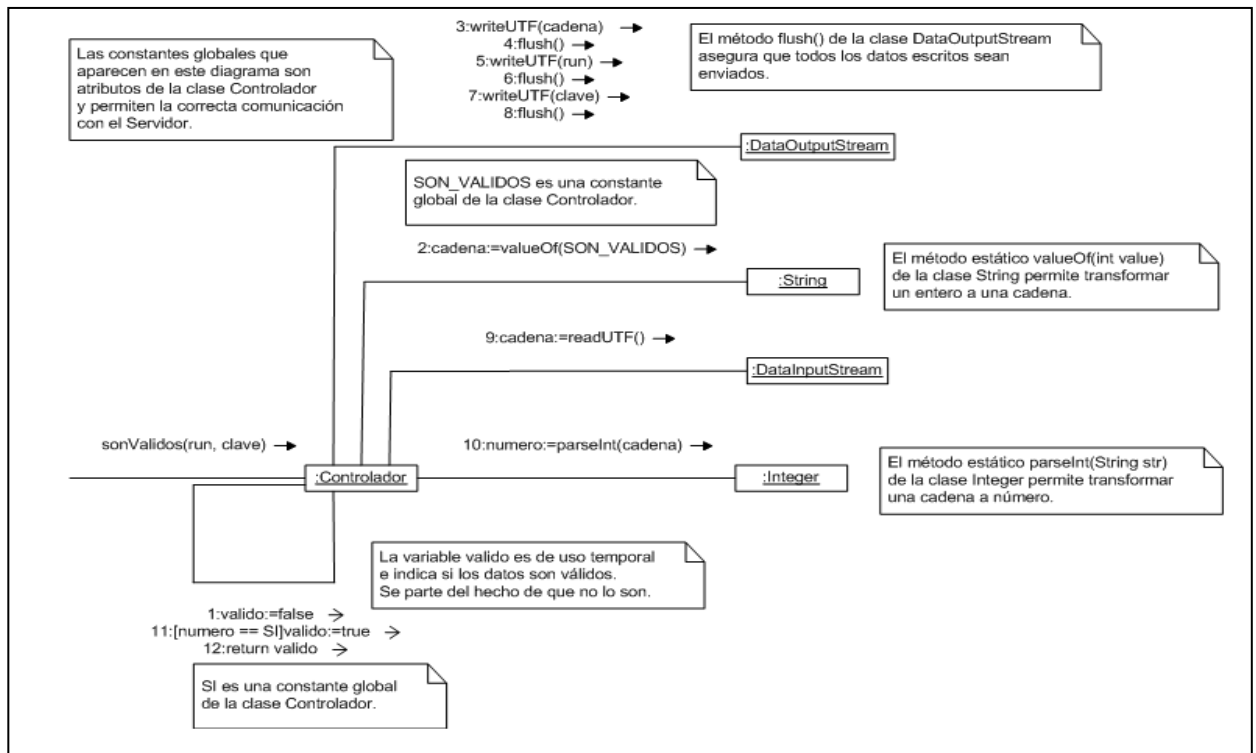


Figura 27, Diagrama sonValidos

Capítulo IV, Desarrollo aplicación nativa sobre Android

La figura N° 28, detalla el diagrama de colaboración *obtenerMedico*, el cual obtiene los datos de un médico desde el servidor pasando por parámetro el run de este.

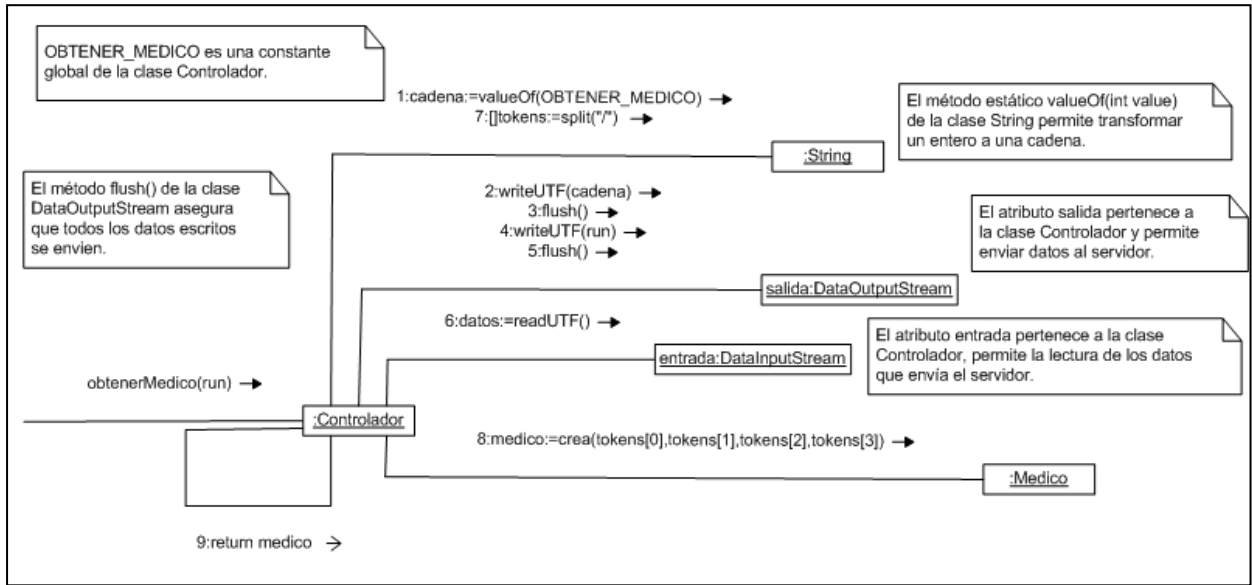


Figura 28, Diagrama obtenerMedico

La figura N° 29, detalla el diagrama de colaboración *cerrarConexion*, en donde el cliente envía al servidor la petición de cerrar la conexión.

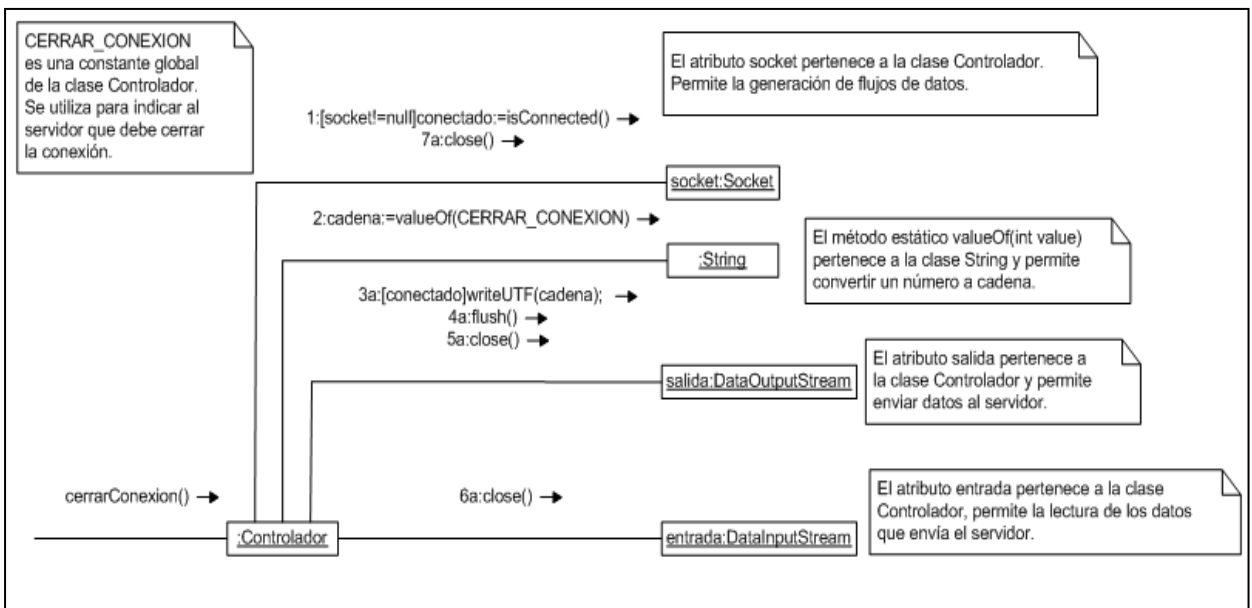


Figura 29, Diagrama cerrarConexion

4.8. DIAGRAMA DE CLASES

El diagrama de clases que se presenta en la figura N° 30, corresponde al paquete persistencia, alojado en el servidor.

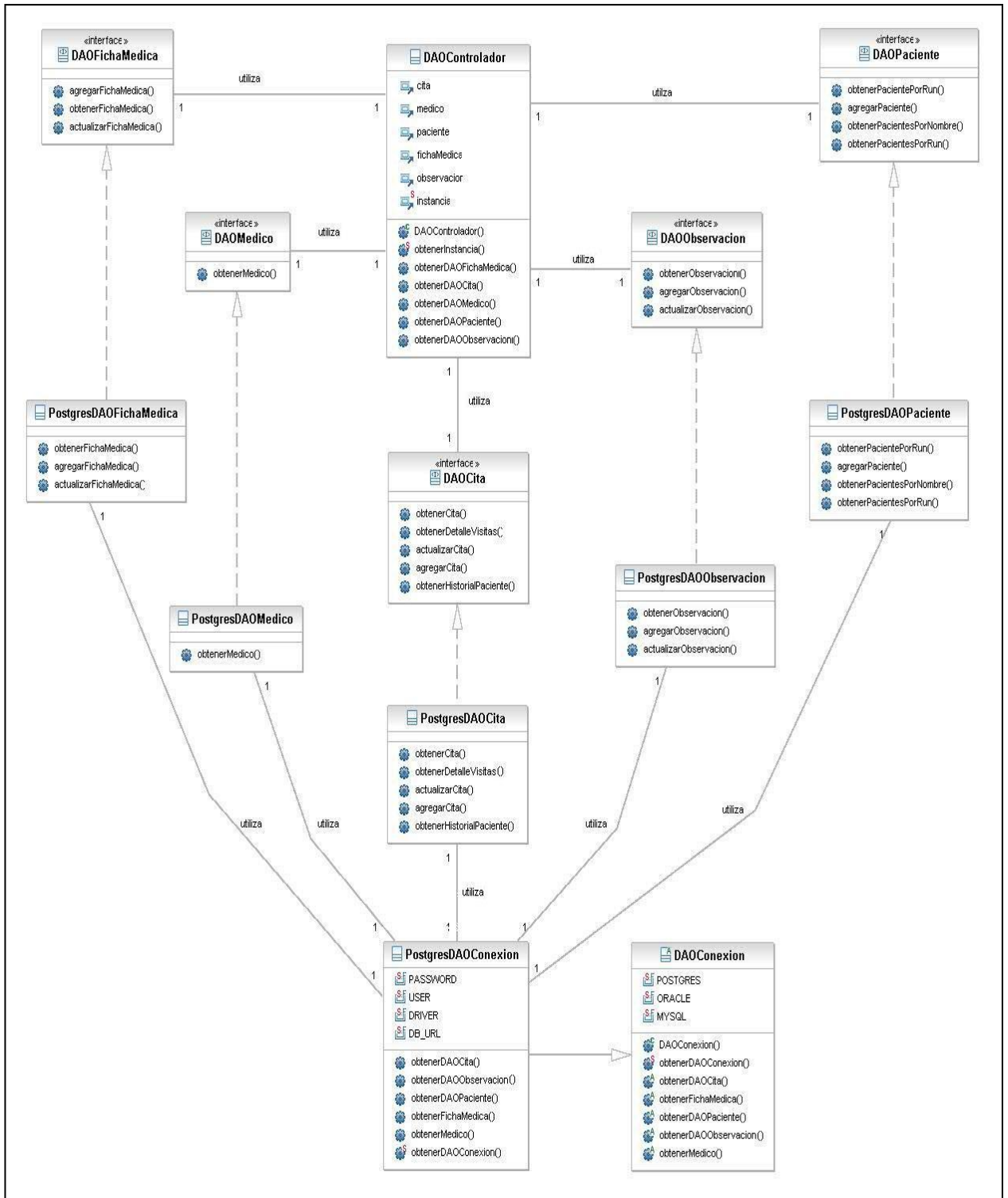


Figura 30, Diagrama de Clases paquete persistencia

Capítulo IV, Desarrollo aplicación nativa sobre Android

A fin de mantener la simplicidad, en el diagrama de clases anterior no se representaron las clases *CitaTO*, *ObservacionTO*, *PacienteTO*, *FichaMedicaTO*, *ObservacionTO* y *MedicoTO*, los cuales forman parte del paquete lógica, estos interactúan con las clases *postgresDAO* correspondientes.

Siguiendo con las clases dispuestas en el servidor, la figura N° 31 representa el paquete comunicación.

Se puede apreciar en la clase *AtenderCliente* atributos con letras mayúsculas, estos corresponden a los servicios ofrecidos por dicha clase y que posteriormente son solicitados por el subsistema alojado en el teléfono.

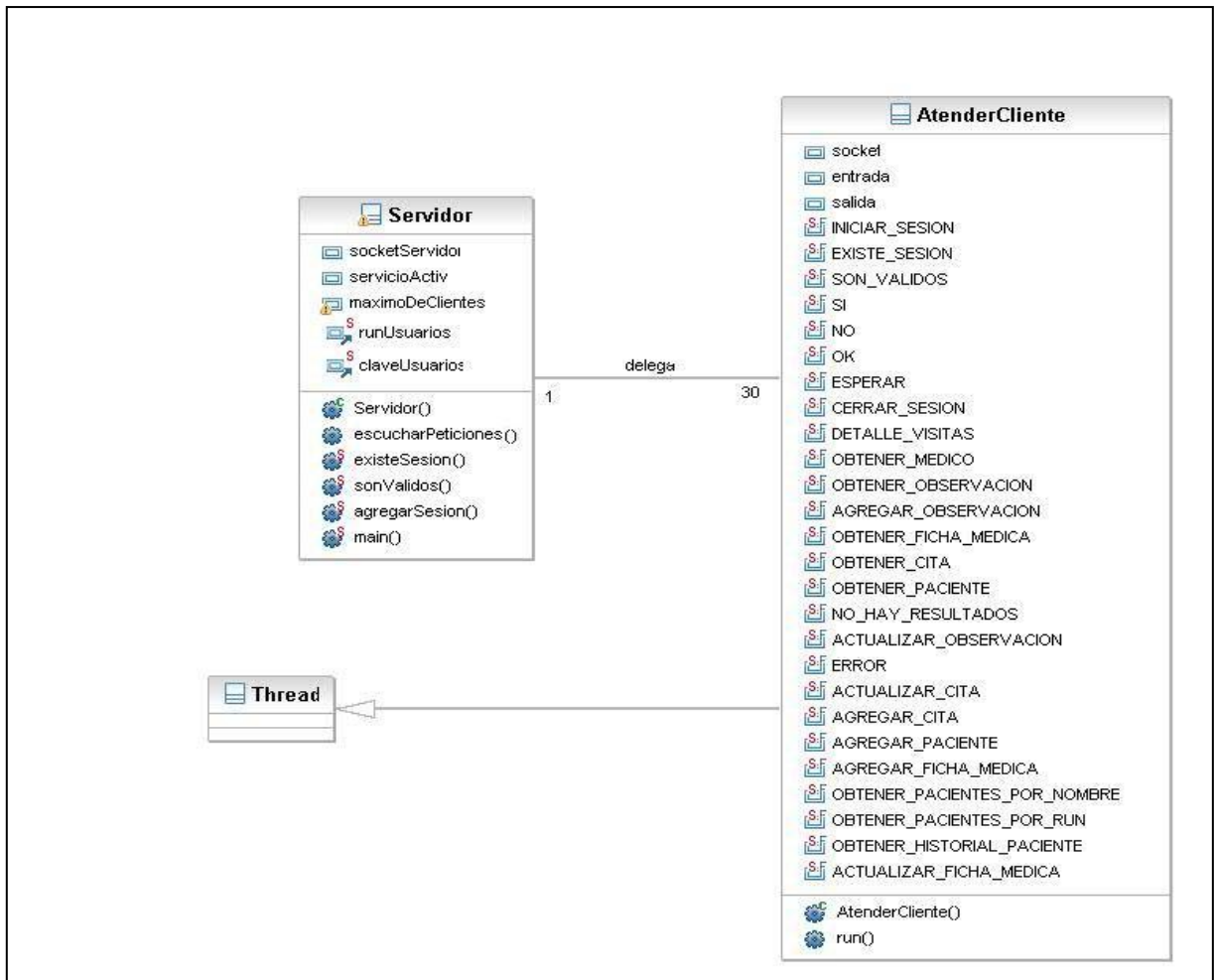


Figura 31, Diagrama de clases paquete comunicación

Capítulo IV, Desarrollo aplicación nativa sobre Android

La figura N° 32, ilustra el diagrama de clases del subsistema agenda médica contenida en el teléfono móvil, con las respectivas vistas, las que se distinguen por tener el método *onCreate()*.

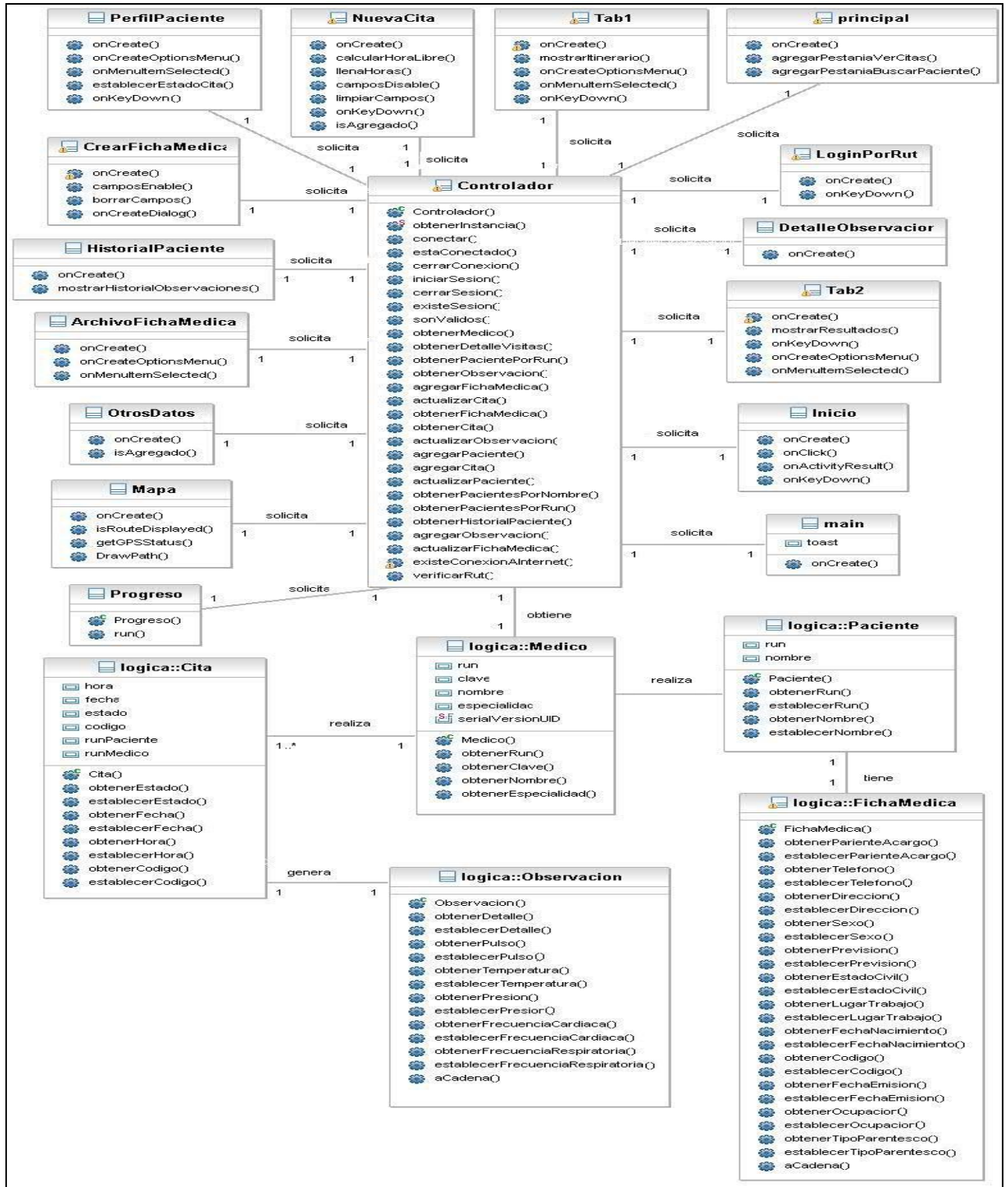


Figura 32, Diagrama de clases agenda medica

4.8. PATRONES DE DISEÑO

Los patrones de diseño entregan solución a problemas que se han repetido constantemente en el tiempo. Para que una solución sea considerada patrón se debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Y además debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias (Pressman, 2002).

El desarrollo de la aplicación tiene considerado la utilización de tres patrones, los cuales se detallarán a continuación.

4.8.1. Patrón Modelo vista controlador (MVC)

Este modelo fue descrito por primera vez en 1979 por Trygve Reenskaug y es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos, los cuales se detallan a continuación.

Modelo: Sirve como representación específica de toda la información con la cual el sistema va a trabajar.

Vista: Presenta el modelo con el que interactúa el usuario, más conocida como interfaz.

Controlador: El controlador responde más bien a eventos, normalmente son acciones que el usuario invoca, implica cambios en el modelo y también en la vista (interfaz).

El uso de este patrón está fuertemente impulsado por Android, ya que esta plataforma utiliza esta modalidad y, al momento de crear un proyecto, por defecto se separan en tres capas.

4.8.2. Singleton

Este patrón de diseño será utilizado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella (Larman, 1999).

Estará implementado en las distintas clases controladoras del proyecto.

4.8.3. Patrón DAO

El objetivo de este patrón es crear una abstracción de la infraestructura de almacenamiento de datos para la aplicación. Una capa de persistencia es útil porque ofrece la funcionalidad de almacenamiento/persistencia de los datos sin revelar detalles específicos de la infraestructura subyacente.

Este patrón será implementado para el acceso y la obtención de datos desde las bases de datos contenidas en el servidor.

4.9. MAPAS DE NAVEGACIÓN

Los mapas de navegación permiten explicar la disposición de las actividades o vistas de una aplicación, de manera que el usuario pueda saber en qué parte se encuentra la utilidad que él requiere y qué debe hacer para llegar a ella.

A continuación, en la figura N° 33 se describe el mapa de navegación correspondiente al subsistema Agenda Médica, en donde cada rectángulo representa una pantalla que en Android se denomina actividad.

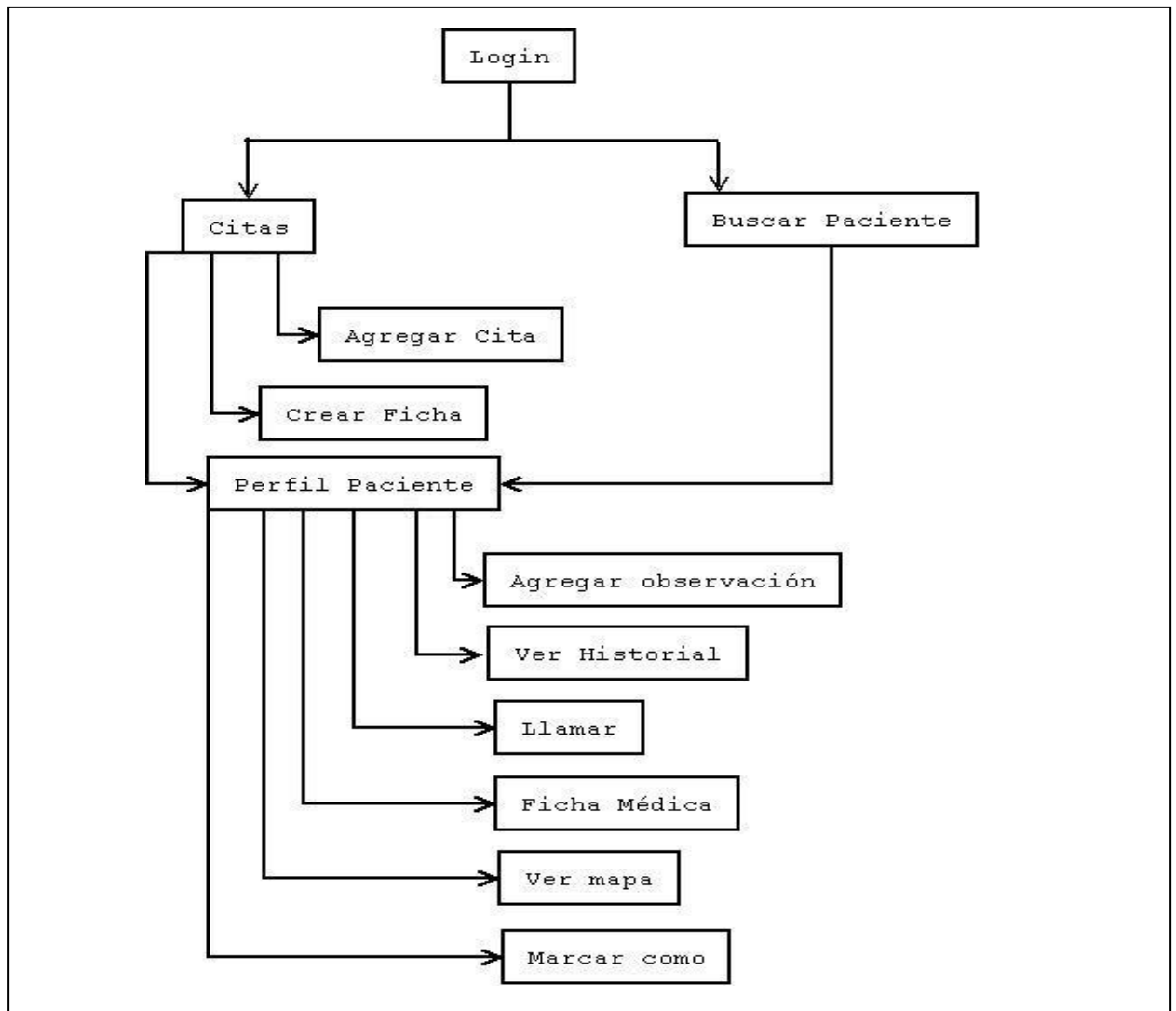


Figura 33, Mapa de navegación

4.9.1. Detalle del Mapa de navegación

Login: En esta parte de la aplicación es donde el usuario debe ingresar los datos pertinentes para identificarse como tal y tener acceso a cada una de las acciones disponibles.

Citas: En esta actividad es donde se tiene acceso a la información de las citas destinadas para el día actual o las futuras. En este detalle se encuentra la hora, el nombre del paciente y la situación actual de la visita (Pendiente, Cancelada, Visitada). Además al seleccionar un paciente se tendrá acceso al detalle de su perfil.

Buscar Paciente: Permite la búsqueda de un paciente existente en los registros, dependiendo de alguno de los parámetros escogidos (Run o Nombre).

Agregar Cita: Situado en el menú de la actividad Citas, es donde se lleva a cabo el registro de una nueva visita a domicilio por parte del médico tratante.

Crear Ficha: Al igual que agregar cita, esta acción se encuentra en el menú de la actividad Citas y permite la creación de una nueva ficha médica.

Perfil Paciente: Contiene el detalle del paciente, y permite agregar nuevas observaciones.

Ver Historial: Ubicado en el menú de perfil paciente, permite ver el historial de visitas y observaciones realizadas al paciente en específico.

Llamar: Ubicado en el menú de perfil paciente, esta actividad permite realizar una llamada al paciente.

Ficha Médica: Ubicado en el menú de perfil paciente, detalla todos los datos ingresados en la ficha médica del paciente.

Ver Mapa: Obtiene la ubicación geográfica, con ayuda de un mapa virtual, de la dirección del paciente.

Marcar Como: Acción que permite cambiar el estado de la visita de un paciente (Pendiente, Cancelado, Visitado).

Capítulo V

IMPLEMENTACIÓN

Y PRUEBAS

Para profundizar el entendimiento del subsistema agenda médica se ha dedicado este apartado a la implementación y pruebas. Se explica cómo fueron desarrollados algunos de los aspectos más importantes de agenda médica, tales como la utilización de mapas, la implementación de la base de datos, diseños de la interfaz y pruebas realizadas a las acciones más importantes de este subsistema

5.1. DESARROLLO DE LA INTERFAZ.

La implementación de las vistas en el caso de Android, como se explicaba en capítulos anteriores, son realizadas en código XML, el cual permite uso de variados componentes y contenedores, para crear distintos diseños dependiendo de las necesidades de una aplicación.

Para entender mejor algunos conceptos básicos, en la figura N° 34 se expone el código de una vista no muy compleja que incluye los componentes que se utilizaron.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical"
  android:background="#FFFFFF">

  <TextView android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Esto es un TextView"
    android:textColor="#000000"></TextView>

  <Button android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Boton"> </Button >

  <EditText android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Esto es un EditText"> </EditText>

  <DatePicker android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:focusableInTouchMode="true"></DatePicker>

</LinearLayout>
```

Figura 34, Código Ejemplo1

El contenedor de esta vista es de tipo *LinearLayout* el cual tiene una orientación horizontal por defecto. Con *android:orientation="vertical"* se está indicando que todos los elementos deben aparecer ordenados por columna, es decir un elemento debajo de otro.

Los demás son widgets que poseen las mismas acciones de los ya conocidos en otros lenguajes. Las propiedades *android:layout_width* y *android:layout_height* son comunes para todos los componentes y en ellas se define el ancho y largo respectivamente que se quiere que ocupe un componente, puede ser “*wrap_content*” el menor espacio posible, o “*fill_parent*” el mayor espacio posible (Burnette, 2008). También se puede en ocasiones especificar un alto (height) y un ancho (width) fijo, de la siguiente forma: *android:layout_width="200px" – android:layout_height="100px"*

Unidades fijas disponibles:

- px (píxeles).
- dip (píxeles independientes del dispositivo)
- sp (píxeles escalados, mejor para tamaños de texto)
- pt (puntos)
- in (inches)
- mm (milímetros)

El código anterior genera la interfaz que se muestra en la figura N°35:



Figura 35, Vista Ejemplo 1

5.2. USO DE MAPAS PARA LA LOCALIZACIÓN

El subsistema agenda médica cuenta con el uso de mapas proporcionados por Google Maps, necesarios para identificar las direcciones de los pacientes y mostrarla de manera gráfica.

El paquete que incluye todas las clases relacionadas con la carga y manejo de mapas directamente desde Google Maps es *com.google.android.maps*. Dentro de este se encuentran las siguientes clases.

MapView: obtiene el mapa solicitado y lo muestra en pantalla.

MapController: gestiona el manejo de un mapa, como desplazamientos o zoom.

GeoPoint: clase que representa un punto geográfico determinado del mapa, según su latitud y longitud.

Overlay: permite dibujar y representar elementos sobre el mapa (marcar direcciones).

MapActivity: es la clase que extiende la clase *Activity*, y permite gestionar mapas.

Para hacer uso de la API anteriormente descrita es necesario registrarse, lo cual se realiza a través de una clave denominada en Android como API key.

5.2.1. Obtención de la API Key

El primer paso a realizar es obtener un resumen MD5 del certificado que se va a usar para firmar la aplicación donde se hará uso de mapas. El resumen puede generarse, con la herramienta *keytool* presente en el SDK de Java. La llamada correspondiente se muestra la figura N° 36.

```
>keytool -list -alias androiddebugkey -keystore  
C:\debug.keystore -storepass android -keypass android  
Huella digital de certificado (MD5):  
0B:41:31:DC:4B:89:22:8E:A5:45:79:C6:13:DA:7E:D4
```

Figura 36, Obtención API Key

Una vez obtenida la huella digital de certificado (MD5), es necesario visitar la Web de registro (Android Maps API key Signup), en donde se ingresa la huella y se aceptan los términos de uso, momento en el cual Google envía un correo a la cuenta Gmail especificada con la API Key. Esta API Key es la que se debe utilizar para obtener y manejar mapas, siendo además única y exclusiva para la aplicación firmada con el certificado utilizado (Rogers, 2009).

El uso de la API Key es la que se muestra en la figura N° 37:

```

1      <?xml version="1.0" encoding="utf-8"?>
2      <LinearLayout
3          xmlns:android="http://schemas.android.com/apk/res/android"
4          android:layout_width="wrap_content"
5          android:layout_height="wrap_content">
6
7          <view class="com.google.android.maps.MapView"
8              android:id="@+id/mapa"
9              android:layout_width="fill_parent"
10             android:layout_height="fill_parent"
11             android:enabled="true"
12             android:clickable="true"
13             android:apiKey="0XfhUsbyT1IblxJanqBiXXTe-jX2WE7YEiaEZrw"/>
14      </LinearLayout>

```

Figura 37, Uso API Key

En la línea número 7 se indica que la vista utilizará mapas, y en la número 13 se le entrega la API Key obtenida en el punto anterior, clave necesaria para el manejo de los mapas en Android.

5.2.2. Implementación de la Actividad que muestra el mapa.

Para hacer uso de los mapas y servicios de Google Maps es necesario que la actividad extienda de *MapActivity*, la cual deriva de la clase *Activity* que incluye las funciones estándares relacionadas con la visión de mapas.

Al mostrar un mapa obtenido desde Google Maps, es necesario tener algunos objetos básicos:

MapView: Obtiene y representa el mapa.

MapController: controla el mapa representado.

MapOverlay: tiene como misión ir actualizando el mapa con diferentes elementos.

En la figura N° 38 se puede observar el uso de los objetos anteriormente descritos que se ha implementado en el subsistema agenda médica.

La línea número 1, indica al mapa que debe contener un controlador de Zoom para acercar o alejar el mapa, la número 3 indica que la vista del mapa no debe ser satelital, debido a que Google maps ofrece dos tipos de vista, las cuales se pueden observar en la figura N° 39.

```

1      mapView.setBuiltInZoomControls(true);
2      mapController = mapView.getController();
3      mapView.setSatellite(false);
4      mapView.setStreetView(true);
    
```

Figura 38, Objetos básicos del mapa



Figura 39, Tipos de vista de Mapa

Habiendo realizado todo lo anterior, es necesario indicar cuál es la zona que se desea mostrar así como el nivel de acercamiento o zoom. Para esto es necesario apoyarse en la clase *GeoPoint*, que representa un punto concreto, una localización determinada a través de sus coordenadas de latitud y longitud. El código que se muestra en la figura N° 40 contiene un ejemplo de lo descrito antes.

```

1 List<address>dirección=geoCoder.getFromLocationName("Madrid 232, chillan, chile", 5);
2 GeoPoint point = new GeoPoint((int) (direccion.get(0).getLatitude() * 1E6), (int)
  (direccion.get(0).getLongitude() * 1E6));
3 mapView.invalidate();
4 mapController.animateTo(point);
5 mapController.setZoom(16);
6 mapController.setCenter(point);

```

Figura 40, Dirección a mostrar en el mapa

La línea 1, convierte la dirección pasada por parámetro en coordenadas representables en el mapa, la siguiente línea las multiplica por un valor para que puedan ser mostradas en el teléfono, *animateTo* indica la posición a mostrar, mientras que *setCenter* centra la vista del mapa fijando como punto centro el *point* a buscar y por último *setZoom* establece el nivel de zoom deseado para el mapa.

Como paso final para generar el mapa, ha de tenerse en cuenta la configuración del *AndroidManifest.xml*, dando los permisos necesarios para el uso de la biblioteca de mapas y e Internet, los cuales se detallan en las figuras N° 41 y 42 respectivamente.

```
<uses-library android:name="com.google.android.maps" />
```

Figura 41, Permiso para uso de la biblioteca Google Maps

```
<uses-permission android:name="android.permission.INTERNET" />
```

Figura 42, Permiso para el uso de Internet

5.2.3. Dibujar una dirección en el mapa.

Como se indicó anteriormente la biblioteca de *com.google.android.maps* permite trabajar con los mapas y servicios ofrecidos desde Google Maps, pero además en ella existe una clase, de nombre *Overlay*, que permite dibujar elementos sobre el mapa.

Es necesario implementar una clase como la que se muestra en la figura N° 43.

```
public class HelloItemizedOverlay extends ItemizedOverlay {
    private ArrayList<OverlayItem> mOverlays = new ArrayList<OverlayItem>();

    public HelloItemizedOverlay(Drawable defaultMarker) {
        super (boundCenterBottom(defaultMarker));
    }

    protected OverlayItem createItem(int i) {
        return mOverlays.get(i);
    }

    public void addOverlay(OverlayItem overlay) {
        mOverlays.add(overlay);
        populate();
    }

    public int size() {
        return mOverlays.size();
    }
}
```

Figura 43, clase *HelloItemizedOverlay*

Ahora solo queda hacer uso de esta clase, desde la actividad que implementa los mapas como se muestra en la figura N° 44, en donde *point* es la coordenada que se desea mostrar.

```
1  itemizedOverlay = new HelloItemizedOverlay(drawable);
2  OverlayItem overlayitem = new OverlayItem(point, "Mapa", "Hola");
3  itemizedOverlay.addOverlay(overlayitem);
4  mapOverlays.add(itemizedOverlay);
```

Figura 44, Implementación *Overlay*

5.3. PRUEBAS DE USABILIDAD.

Las pruebas de usabilidad tienen como objetivo estudiar y medir el grado de facilidad con que los usuarios interactúan con algún objeto creado, ya sea una herramienta, máquina o, en el caso de este proyecto, una aplicación.

Existen distintas formas de medir la usabilidad y son las que se describen a continuación:

Expertos: Consiste en encargar a un experto que evalúe (evaluación por criterios o heurística) la aplicación. Existen muchos profesionales independientes y empresas que se dedican a esta tarea, lo cual obviamente implica incurrir en costos monetarios.

Encuestas: Otra de las opciones es realizar encuestas para comprobar la usabilidad de la aplicación. La encuesta debería ser diseñada y realizada sobre usuarios actuales o potenciales.

Pruebas de usabilidad: Consiste en contar con unos 5 usuarios en una sala, los cuales realizarán una navegación "asistida" en donde se les solicitará que lleven a cabo las tareas para las cuales la aplicación fue diseñada, en tanto el equipo de diseño, desarrollo y otros involucrados toman nota de la interacción, particularmente de los errores y dificultades con las que se encuentren los usuarios.

5.3.1. Acciones a medir

Las acciones que se van a solicitar al usuario en las pruebas de usabilidad son las que se describen a continuación:

- 1.- Identifíquese en el sistema, mediante el campo solicitado.
- 2.- Vea las citas de días anteriores.
- 3.- Vea el detalle de un paciente.
- 4.- Busque un paciente en particular.
- 5.- Agregue una nueva cita médica.
- 6.- Cree una nueva ficha médica.
- 7.- Cambie el estado de una cita médica.
- 8.- Vea el mapa con la ubicación de la dirección del paciente.
- 9.- Llame a un paciente.
- 10.- Vea detalle del historial de observaciones.

El formato de la prueba de usabilidad aplicada a un grupo de personas elegidas al azar es la que se puede apreciar en el Anexo E.

5.3.2. Resultados de las pruebas de usabilidad

Para llevar a cabo las pruebas de usabilidad, fue necesario contar con el apoyo de diez usuarios ajenos al equipo de desarrollo, quienes realizaron el conjunto de actividades establecido en el punto anterior.

La figura N° 45 muestra algunos de los usuarios que utilizaron y evaluaron el subsistema agenda médica móvil.

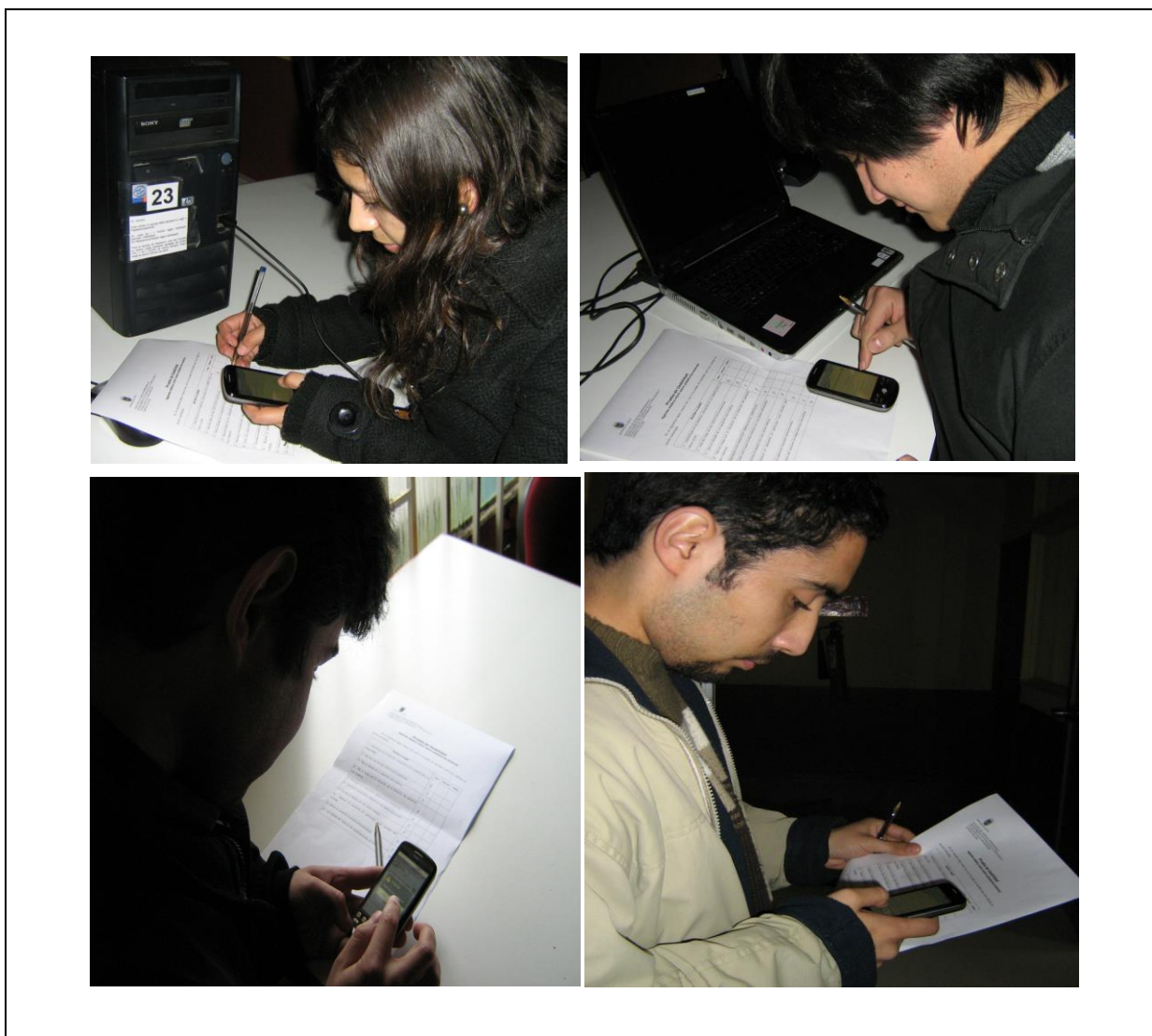


Figura 45, Aplicación prueba de usabilidad

Capítulo V, Implementación y pruebas

Por cada actividad, el usuario tuvo que seleccionar el nivel de dificultad que experimentó al realizar la tarea. Los niveles a elección son; Fácil, Normal y Difícil, siendo la última una respuesta de especial cuidado señal de un potencial error de diseño.

Cada nivel de elección tiene una puntuación de 0, 1 y 2, para Difícil, Normal y Fácil, respectivamente. Por lo tanto, entre mayor sea la puntuación que tenga un ítem, mayor será el nivel de usabilidad.

Los porcentajes obtenidos en las encuestas son las presentadas en la Tabla Nº 19.

Tabla 19, Resultado prueba de usabilidad

Actividad a medir	Fácil	Normal	Difícil
1.- Identifíquese en el sistema, utilizando el Rut 164972135 y la clave 1234.	100%	0%	0%
2.- Vea las citas de día 19/07/2010.	95%	5%	0%
3.- Vea el perfil del paciente Raúl Medina.	90%	10%	0%
4.- Vea el mapa con la ubicación de la dirección del paciente Raúl Medina.	70%	25%	5%
5.- Agregue una nueva Cita médica al Rut 154972135.	80%	20%	0%
6.- Cree una nueva ficha médica con sus propios datos personales.	80%	20%	0%
7.- Cambie el estado de una visita médica.	95%	5%	0%
8.- Busque un paciente cuyo nombre sea Andrea.	100%	0%	0%
9.- Llame al paciente buscado anteriormente.	85%	15%	0%
10.- Vea detalle del historial de observaciones de Andrea.	95%	5%	0%

Como se puede observar en la tabla la totalidad de los ítems tuvo buenas evaluaciones (sobre el 70%), lo que indica que el nivel de usabilidad es aceptable, y a los usuarios se les hizo fácil aprender a utilizar la aplicación.

5.4. PRUEBAS DE UNIDAD, DESEMPEÑO Y TENSIÓN

Para entregar un software de calidad, es necesario asegurar que cada uno de los componentes funciona correctamente, de tal manera que el usuario no se encuentre con sorpresas inesperadas al momento de utilizar la aplicación. Para esto existen pruebas que permiten al equipo de desarrollo asegurar que cada una de las funcionalidades ofrecidas cumpla debidamente con lo requerido.

5.4.1. Pruebas de unidad

Se ha desarrollado un tipo de prueba de unidad para este apartado, las de caja negra, en donde se prescindir de los detalles del código y se limita a lo que se ve desde el exterior.

Pruebas de caja Negra

En la tabla N° 20, se detalla la prueba de caja negra "Autenticar usuario".

Caso de Prueba de Respuesta	
Código Caso de Prueba: 1.0	Código Caso de Uso: Autenticar usuario.
Descripción de la Prueba: Esta prueba evaluará el ingreso de los datos de inicio de sesión.	
Condiciones de Ejecución: No tiene.	
Entrada / Pasos de Ejecución: <ol style="list-style-type: none"> 1. Se selecciona la opción de autenticación. 2. El usuario ingresa el dato requerido para la identificación, ya sea código QR, o Rut y clave. 	
Resultado Esperado: <ol style="list-style-type: none"> 1. El sistema ingresa a la aplicación y muestra la vista principal con los datos de las citas. 	
Evaluación de la Prueba: Ok	

Tabla 20, Prueba de caja negra Autenticar Usuario

Capítulo V, Implementación y pruebas

En la tabla N° 21, se detalla la prueba de caja negra "Crear ficha médica".

Caso de Prueba de Respuesta	
Código Caso de Prueba: 1.0	Código Caso de Uso: Crear ficha médica
Descripción de la Prueba: Esta prueba evaluará el ingreso de una ficha médica al sistema, con alguno de los campos obligatorios faltantes	
Condiciones de Ejecución: El médico debe estar autenticado en el sistema.	
Entrada / Pasos de Ejecución: <ol style="list-style-type: none"> 1. Se ingresa a la opción crear ficha médica 2. Se completan algunos de los campos marcados con (*), puesto que estos son obligatorios 3. Se selecciona la opción crear ficha. 	
Resultado Esperado: <ol style="list-style-type: none"> 1.- Se indicará que faltan ingresar todos los campos obligatorios no ingresados. 	
Evaluación de la Prueba: Ok	

Tabla 21, Prueba de caja negra Crear ficha médica

En la tabla N° 22, se detalla la prueba de caja negra "Ver dirección en el mapa".

Caso de Prueba de Respuesta	
Código Caso de Prueba: 1.0	Código Caso de Uso: Ver dirección en el mapa
Descripción de la Prueba: Esta prueba evaluará la búsqueda de una dirección correcta en el mapa.	
Condiciones de Ejecución: <ol style="list-style-type: none"> 1.- El médico debe estar autenticado en el sistema. 2.- El médico se debe encontrar en el perfil del paciente. 3.- El paciente debe tener en el perfil su dirección. 	
Entrada / Pasos de Ejecución: <ol style="list-style-type: none"> 1.- Se selecciona la opción ver mapa, ubicada en el menú del perfil paciente 	
Resultado Esperado: <ol style="list-style-type: none"> 1.- Se carga el mapa mostrando el camino a seguir entre la ubicación actual del médico y la dirección del paciente. 	
Evaluación de la Prueba: Ok	

Tabla 22, Prueba de caja negra ver dirección en el mapa

Capítulo V, Implementación y pruebas

En la tabla N° 23, se detalla la prueba de caja negra “Buscar visitas por fecha”.

Caso de Prueba de Respuesta	
Código Caso de Prueba: 1.0	Código Caso de Uso: Buscar visitas por fecha
Descripción de la Prueba: Esta prueba evaluará la búsqueda de visitas en un día que tenga visitas agendadas.	
Condiciones de Ejecución: 1.- El médico debe estar autenticado en el sistema. 2.- Debe encontrarse en la pestaña citas.	
Entrada / Pasos de Ejecución: 1.- El médico ingresa la fecha 10/07/2010.	
Resultado Esperado: 1.- Se actualiza la lista de visitas mostrando todas las visitas agendadas para el día seleccionado.	
Evaluación de la Prueba: Ok	

Tabla 23, Prueba de caja negra buscar visitas por fecha

En la tabla N° 24, se detalla la prueba de caja negra “Agregar visita médica”.

Caso de Prueba de Respuesta	
Código Caso de Prueba: 1.0	Código Caso de Uso: Agregar visita médica
Descripción de la Prueba: Esta prueba evaluará el ingreso de una nueva visita médica a la agenda.	
Condiciones de Ejecución: 1.- El médico debe estar autenticado en el sistema. 2.- El médico debe encontrarse en la pestaña citas.	
Entrada / Pasos de Ejecución: 1.- El médico debe seleccionar la opción agregar cita , ubicada en el menú de la pestaña citas. 2.- El médico debe ingresar todos los datos marcados como obligatorios.	
Resultado Esperado: 1.- La aplicación guarda los datos de la cita y avisa el éxito de la operación.	
Evaluación de la Prueba: Ok	

Tabla 24, Prueba de caja negra Agregar visita médica

5.4.2. Pruebas de desempeño y tensión

Las pruebas de tensión se realizaron midiendo el tiempo que se tomó en llevar a cabo la consulta e ingreso de distintas cantidades de registros (10, 100, 1.000 y 10.000) y disponerlos o dar aviso en la pantalla del teléfono móvil. Por cada cantidad se realizaron ocho pruebas, eliminando el mejor y peor resultado, para luego obtener el tiempo promedio en cada módulo analizado, los cuales se representan en los gráficos siguientes por medio de puntos rojos.

La cantidad de registros utilizados para la evaluación es una estimación realizada en base a los antecedentes del Consultorio Federico Puga Borne, que hoy en día cuenta con más de 1.600 registros de pacientes.

La prueba que se detalla en la figura N° 46 corresponde a la búsqueda y despliegue de citas para una fecha determinada.

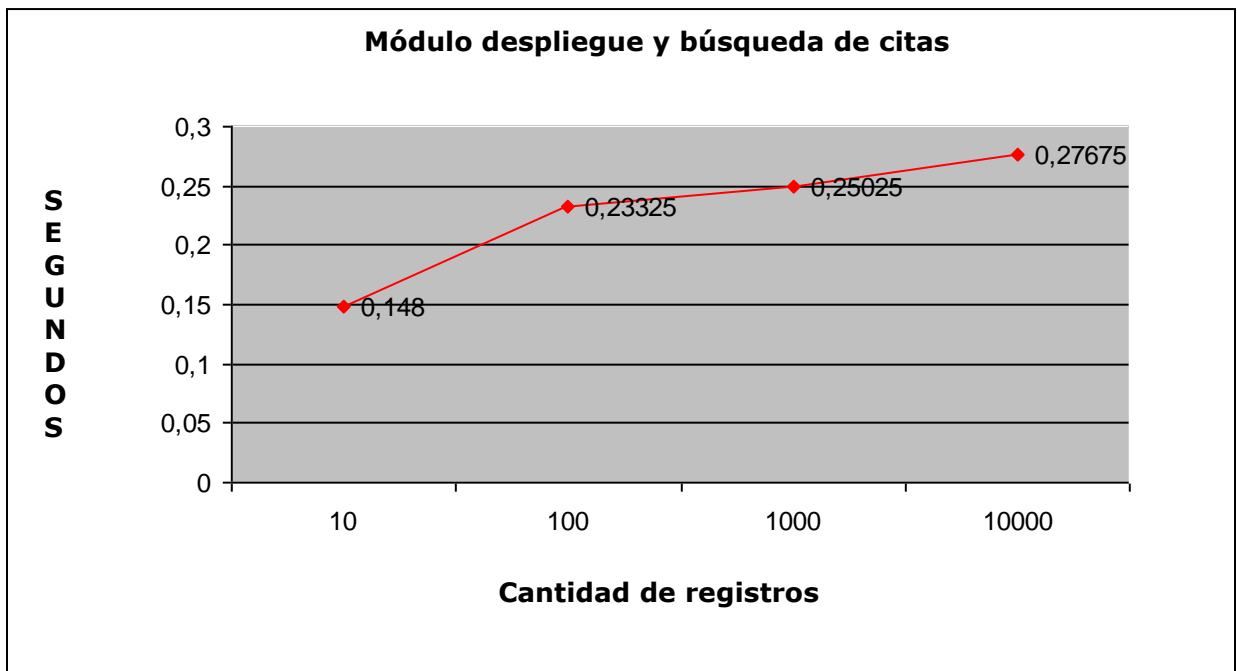


Figura 46, Módulo despliegue y búsqueda

La figura N° 47, detalla el gráfico de los tiempos obtenidos al autenticar a un médico mediante un run y su clave entre las distintas cantidades de registros.

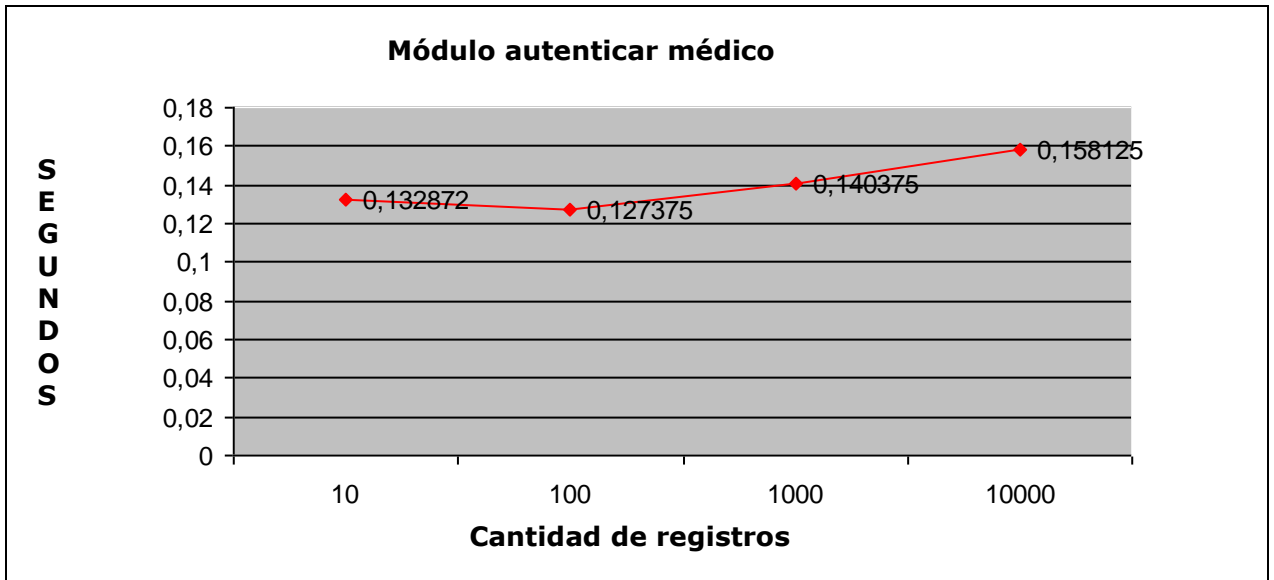


Figura 47, Módulo autenticar medico

La figura N° 48, detalla el gráfico con los tiempos obtenidos al ingresar visitas médicas en la agenda.

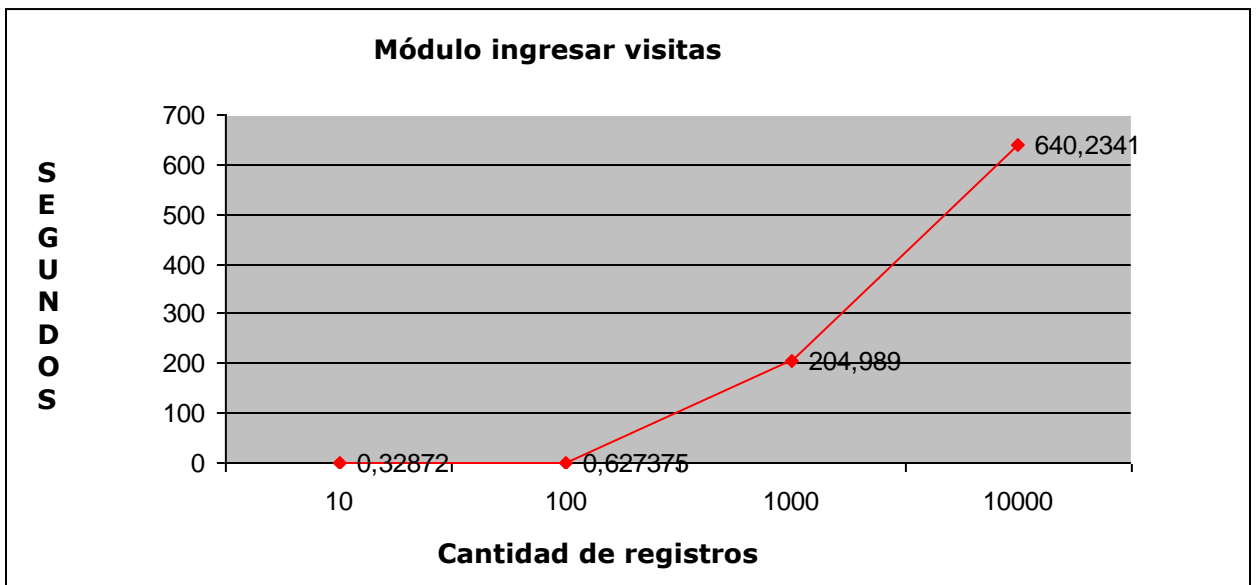


Figura 48, Módulo ingresar visita

Al término de las pruebas se pudo demostrar que el tiempo de respuesta estimado como máximo en la tabla de requisitos no funcionales (tabla Nº 1) era 7 segundos, el cual no se cumple cuando se ingresan más de 10.000 registros en la base de datos.

Cabe destacar además, que se realizaron pruebas para medir el tiempo de respuesta a cada una de las funciones de la plantilla combinada del capítulo IV (tabla Nº 3), y los resultados obtenidos fueron todos bajo el tiempo establecido como máximo.

Las pruebas fueron realizadas haciendo uso de un teléfono HTC magic con las características descritas en el apartado 2.2.3.1 del capítulo II, y se hizo uso de un chip telefónico de la empresa ENTEL PCS con un plan de datos de Banda Ancha Móvil de 700 Kbps, para la conexión a Internet.

5.5. DETALLES DE LA PERSISTENCIA DE DATOS

Para lograr la persistencia de los datos que la agenda médica requiere, se ha recurrido a la implementación de una base de datos contenida en un servidor Web con la finalidad de que el teléfono celular se conecte a ella y de esta forma acceda a los datos requeridos por el médico.

Se han necesitado 5 tablas para la implementación de la base de datos, las cuales son las que se especifican a continuación y se pueden apreciar gráficamente en el modelo entidad relación (figura Nº 19):

Paciente: contiene los principales datos del paciente.

Médico: contiene los datos de los médicos.

Observación: contiene los datos de las observaciones realizadas por los médicos a los pacientes.

Cita: Contiene el detalle de las visitas médicas.

Ficha médica: posee toda la información de una ficha médica de los pacientes ingresados en el sistema.

5.5.2. Detalle de tablas

La tabla N° 25, corresponde a la tabla "paciente" que permite almacenar los datos principales de los pacientes.

Tabla paciente		
Nombre del atributo	Descripción	Tipo
run	Es la clave primaria de la tabla y corresponde al run del paciente.	Character Varying
nombre	Nombre completo del paciente que incluye nombres y apellidos.	Character Varying

Tabla 25, Tabla Paciente

En la tabla N° 26, que se describe a continuación, se detallan los datos que posee la tabla "medico".

Tabla medico		
Nombre del atributo	Descripción	Tipo
run	Es la clave primaria de la tabla y corresponde al run del médico.	Character Varying
nombre	Nombre completo del médico que incluye nombres y apellidos.	Character Varying
clave	Corresponde a la clave de acceso del médico.	Character Varying
especialidad	Es la especialidad en la que se desempeña el médico.	Character Varying

Tabla 26, Tabla Médico

Capítulo V, Implementación y pruebas

En la tabla N° 27, que se describe a continuación, se detallan los datos que posee la tabla "cita".

Tabla cita		
Nombre del atributo	Descripción	Tipo
codigo	Es la clave primaria de la tabla y corresponde al código de la visita.	integer
fecha	Corresponde a la fecha en la cual se desea crear la visita médica.	Character Varying
estado	Corresponde al estado actual de la visita, la cual puede ser "Visitada, pendiente, cancelada".	Character Varying
hora	Corresponde a la hora en la que se fijará la visita médica.	Character Varying
run_medico	Clave foránea proveniente de la tabla médico y corresponde al run de un médico.	Character Varying
run_paciente	Clave foránea proveniente de la tabla paciente y corresponde al run de un paciente.	Character Varying

Tabla 27, Tabla Cita

En la tabla N° 28, que se presenta a continuación, se detallan los datos que posee la tabla "observacion".

Tabla observacion		
Nombre del atributo	Descripción	Tipo
detalle	Corresponde al detalle general que realizan los médicos cuando hacen una visita.	Character Varying
codigo_ficha	(Clave foránea) Asocia una observación con la ficha médica del paciente.	integer
run_medico	(Clave foránea) Corresponde al run del médico autor de la observación.	Character Varying
codigo_cita	Es una clave foránea proveniente de la tabla Cita y corresponde al código de esta.	Character Varying
pulso	Dato que corresponde al pulso del paciente.	Character Varying
temperatura	Dato que corresponde a la temperatura del paciente.	Character Varying
presion	Dato que corresponde a la presión del paciente.	Character Varying
fRespiratoria	Dato que corresponde a la frecuencia respiratoria del paciente.	Character Varying
fCardiaca	Corresponde a la frecuencia cardiaca del paciente	Character Varying

Tabla 28, Tabla observacion

En la tabla N° 29, que se presenta a continuación, se detallan los datos que posee la tabla "fichamedica".

Tabla fichamedica		
Nombre del atributo	Descripción	Tipo
codigo	Clave primaria de la tabla, y corresponde al código de identificación.	Serial
parienteAcargo	Corresponde al nombre completo del pariente que esta a cargo del paciente.	integer
Telefono	Teléfono de comunicación con el paciente	Character Varying
direccion	Lugar de residencia del paciente.	Character Varying
sexo	Sexualidad del paciente.	Character Varying
prevision	Tipo de previsión con la que cuenta el paciente (Isapre, Fonasa, Sin previsión).	Character Varying
run_paciente	Es una clave foránea proveniente de la tabla paciente y corresponde al run de este.	Character Varying
estado_civil	Estado civil en el que se encuentra el paciente.	Character Varying
lugar_trabajo	Lugar en donde el paciente se desempeña laboralmente.	Character Varying
fecha_nacimiento	Dato que corresponde a la fecha de nacimiento del paciente con formato día/mes/año.	Character Varying
fecha_emision	Dato que corresponde a la fecha en la que se creó la ficha médica posee el mismo formato que la fecha de nacimiento.	Character Varying
ocupacion	Indica a lo que se dedica el paciente (Ej: Estudiante, empleado).	Character Varying
tipo_parentesco	Tipo de parentesco que tiene el pariente a cargo del paciente a tratar.	Character Varying

Tabla 29, Tabla fichamedica

5.6. SEGURIDAD DE LA APLICACIÓN

La seguridad en una aplicación es necesaria para evitar que personas ajenas tengan acceso a información privada, debido a este motivo, la aplicación agenda médica cuenta con un inicio de sesión a la que sólo podrán ingresar usuarios autorizados quienes serán identificados mediante un **código QR personal e intransferible que contendrá el run del médico y su clave**. También se ofrece la posibilidad de ingresar a la aplicación proporcionando solamente el run y la clave de manera manual en el caso de no contar con el código QR.

Toda la información que maneja el sistema está almacenada en una base de datos alojada en un servidor el cual protege esta base con una contraseña para que solo el administrador responsable haga uso y las modificaciones pertinentes asegurando de esta forma la integridad de los datos.

5.6.1. Código QR.

El código QR (Quick Response Barcode) fue creado en Japón en 1994 por la compañía japonesa Denso-Wave, como una alternativa más rápida y con la posibilidad de tener más contenido que los tradicionales códigos de barras, ya que almacenan información en una matriz de puntos o un código de barras bidimensional. La figura N° 49 corresponde a un código QR.



Figura 49, Código QR

El uso del código QR está muy extendido en los dispositivos móviles o cualquier otro dispositivo que disponga de una cámara, entonces, para seguir la tendencia y gracias a la rapidez y facilidad que ofrece para capturar datos e información ha sido elegida para el inicio de sesión.

5.6.2. Implementación del inicio de sesión.

Para realizar la lectura del código QR es necesario tener instalado en el celular una aplicación gratuita que ofrece este servicio llamada Barcode Scanner. Al hacer un llamado a esta aplicación (figura N° 50) Barcode tomará el control mientras realiza el scaneo y retornará el mando a la aplicación agenda médica cuando identifique el contenido del código QR, operación que no tarda más de 2 segundos.

El código que realiza el llamado a la aplicación Barcode es el siguiente:

```

1  Intent intent = new Intent("com.google.zxing.client.android.SCAN");
2  intent.putExtra("SCAN_MODE","QR_CODE_MODE");
3  this.startActivityForResult(intent, 1);

```

Figura 50, Llamado a Barcode Scanner

En la línea número 1, es donde se hace el llamado al Barcode Scanner, la siguiente línea selecciona el modo de Scaneo, en este caso QR.

Como el código QR contiene información a descifrar, es necesario obtener el resultado de la operación anterior, lo cual se lleva a cabo implementando un método que espere el resultado de la lectura, el que se describe en la figura N° 51.

```

1  protected void onActivityResult(int requestCode,int resultCode,Intent data) {
2      super.onActivityResult(requestCode, resultCode, data);
3      if(resultCode == RESULT_OK){
4          String texto = data.getStringExtra("SCAN_RESULT");
5          Intent intento= new Intent(main.this,principal.class);
6          intento.putExtra("QR", data.getStringExtra("SCAN_RESULT"));
7          startActivity(intento);
8      }
9  }

```

Figura 51, Resultado QR

En la línea número 4, es donde se obtiene el texto de la lectura realizada por Barcode Scanner, para luego realizar con dicho texto lo que se desee, en este caso pasar a la siguiente actividad.

5.6.3. Generación del código QR.

Para la generación del código QR se hará uso de la página "http://qrcode.kaywa.com", la cual ofrece gratuitamente la generación de códigos QR. En esta página el encargado de la aplicación **deberá generar el código, ingresando el Rut del médico y una clave seleccionada, separados por un guión (-)**, tal y como se muestra en la figura N° 52.

Este código debe ser entregado al médico para que él pueda identificarse en la aplicación agenda médica.

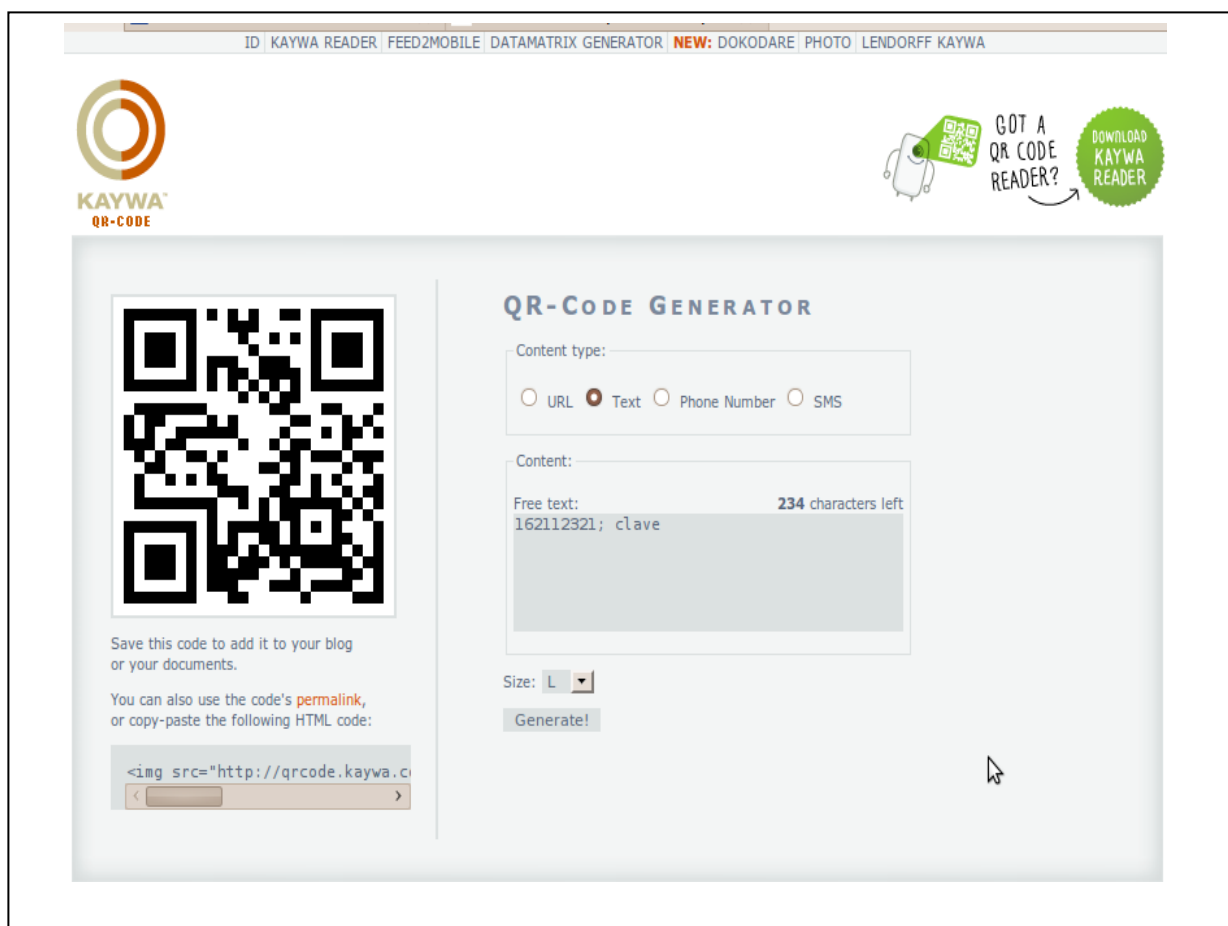


Figura 52, Página generadora de códigos QR

Capítulo VI

INTERACCIÓN CON EL SERVIDOR WEB

El almacenamiento de los datos requeridos para el correcto funcionamiento de la agenda médica se ha realizado en una base de datos contenida en un servidor Web, el cual interactúa por medio de sockets con la aplicación del dispositivo móvil.

En este capítulo se explica el detalle de lo anterior a fin de que el lector comprenda cómo el teléfono y el servidor se pueden comunicar entre sí.

5.1. SOCKET PARA LA INTERACCIÓN

El proyecto realizado se basa en la arquitectura cliente/servidor (figura Nº 21 del capítulo IV) por lo tanto fue necesario implementar una forma de comunicación entre el subsistema alojado en el teléfono y la del servidor. La herramienta utilizada para dicho procedimiento es el socket, el que permite comunicar e intercambiar cualquier datos, generalmente de manera fiable y ordenada entre los dos subsistemas. Los sockets se crean y se utilizan con un sistema de peticiones o de llamadas de función.

Android hace uso de ellos para la comunicación entre la aplicación del teléfono celular y la base de datos contenida en el servidor, es por esta razón que ha sido implementado en el desarrollo de la interacción.

5.2. IMPLEMENTACIÓN DE LOS SOCKETS

Para la implementación de los sockets es necesario (en la máquina que actúa como servidor) habilitar un puerto en común para que los dos subsistemas se comuniquen entre si, en el caso del proyecto se habilitó el puerto 6500 (pudiendo ser cualquier otro) para escuchar la petición de los clientes. La aplicación que actúa como servidor fue escrita en el lenguaje java, esta se encarga de atender las peticiones provenientes de la agenda médica contenida en el teléfono.

Se ha establecido que la aplicación que actúa como servidor puede satisfacer a una demanda máxima de 30 clientes en forma simultánea, debido a que en el consultorio que se utilizó como base para la obtención de información "Federico Puga Borne", cuenta hoy en día con 10 médicos que realizan visitas en terreno. Por lo tanto se ha dejado una holgura en el caso que se sumen nuevos especialistas que ofrezcan el servicio. (La holgura fue realizada al azar).

En la figura N° 53, se detalla la inicialización del servidor, en donde se determina que la atención máxima de clientes será 30, y crea el *ServerSocket* con el puerto habilitado para escuchar.

```
public Servidor() {
    maximoDeClientes    = 30;
    servicioActivo      = true;
    misClientes         = new Vector<Socket>();
    try{
        socketServidor  = new ServerSocket(6500);
    }catch(Exception e){
        e.printStackTrace();
    }//fin catch
} //fin constructor
```

Figura 53, Inicializa socket

Cuando un usuario se conecta al servidor, este último crea un socket por el cual ambos (cliente y servidor) se comunicarán y se lo entrega a la clase *AtenderCliente*, la cual se encarga de responder las peticiones provenientes del teléfono (cliente). La forma en que el socket se crea es la que se detalla en la figura N° 54:

```
public AtenderCliente(Socket s){
    try{socket = s;
        entrada = new DataInputStream(socket.getInputStream());
        salida = new DataOutputStream(socket.getOutputStream());
    }catch(Exception e){
        e.printStackTrace();
    }//fin catch
} //fin constructor
```

Figura 54, Crear Socket

La figura N° 55, detalla la forma en que el servidor escucha las peticiones, esto se realiza siempre y cuando el servicio se encuentre activo y la cantidad de clientes conectados al servidor no supere los 30.

```
1      if(misClientes.size() < maximoDeClientes){
2          System.out.println("- Esperando alguna peticion.");
3          Socket socketCliente = socketServidor.accept();
4          System.out.println("- Peticion recibida.");
5          misClientes.add(socketCliente);
6          AtenderCliente ac = new AtenderCliente(socketCliente);
7          ac.start();
8      }
```

Figura 55, Recibe peticiones

Cuando escucha alguna petición la acepta (línea 3) y agrega el socket del cliente que hizo la petición al vector de clientes y llama a la clase *AtenderCliente* la cual es un *Thread* y que está encargada de responder cada una de las peticiones realizadas por este socket.

La forma en que la clase *AtenderCliente*, realiza una operación es consultando por el tipo de petición que hizo el cliente y dependiendo de esta responde lo solicitado, la figura N° 56, detalla una petición de validación.

```
1      if(peticion.equals(VALIDAR)){
2          //Se lee el dato a validar.
3          String run = entrada.readUTF();
4          boolean esValido = esValido(run);
5          salida.writeUTF(String.valueOf(esValido));
6          salida.flush();
7      }
```

Figura 56, Atender petición

En la línea 3, se leen los datos enviados por el cliente y se llama al método “*esValido*” el cual está encargado de realizar la consulta a la base de datos, para luego en la línea 5 devolver el resultado de la consulta.

Siguiendo el ejemplo anterior, en la figura N° 57 se detalla cómo el cliente hace la petición al servidor de Validar al usuario.

```
1      public boolean esValido(String run){
2      try{//Se indica al servidor que operacion debe realizar.
3          salida.writeUTF("VALIDAR");
4          salida.flush();
5          //Envio del run.
6          salida.writeUTF(run);
7          salida.flush();
8          //Lectura de la respuesta.
9          String esValido = entrada.readUTF();
10         return Boolean.parseBoolean(esValido);
11     }catch(Exception e){
12         e.printStackTrace();
13     }//fin catch
14     return false;
15 }//fin
```

Figura 57, Petición al servidor

En la línea 3 se escribe por el socket informando al servidor que necesita Validar un dato, luego el cliente escribe el run a validar y se queda esperando la respuesta. En la línea 9 se lee la respuesta entregada por el servidor el cual devuelve el resultado de la operación.

CONCLUSIONES

Tras el desarrollo del proyecto, es clave destacar cómo las tecnologías móviles se adaptan y permiten potenciar el mundo laboral. Su evolución en la actualidad ha llegado a tal punto que ya es posible contar con servicios que hasta hace un tiempo eran imposibles de imaginar.

A continuación se presentan las conclusiones específicas del proyecto que buscan relatar brevemente los aspectos de importancia:

- En primera instancia es conveniente indicar que el proyecto se originó como una iniciativa que busca dejar en evidencia las bondades de las tecnologías móviles, y que a su vez sirva de futura referencia a otros proyectos que deseen profundizar en el tema. Bajo esta perspectiva, se optó por utilizar Android, una plataforma creada recientemente por Google que inicia sus primeros pasos en el mercado de las aplicaciones para SmartPhone.
- Fue de vital importancia la participación del Consultorio Federico Puga Borne cuyo personal a cargo manifestó en todo momento un profundo interés por cooperar con el proyecto. En los primeros capítulos relacionados con el análisis y diseño jugaron un rol fundamental técnicas como la entrevista, visitas a terreno, encuestas y, por supuesto, la metodología de trabajo utilizada (Iterativa Incremental) que aseguró una constante comunicación con el usuario final evitando tomar rumbos equivocados.
- Un aspecto interesante a mencionar es el cómo varían los factores críticos en una aplicación dependiendo de la tecnología utilizada para su desarrollo, en este caso, el concepto más innovador fue el de usabilidad que requirió la ejecución de pruebas específicas para medir su nivel de presencia con resultados claramente positivos.
- Gran parte de las funcionalidades de Android se basan en procedimientos escritos en java, un lenguaje de alto nivel que permite representar fielmente la lógica del negocio gracias a su filosofía orientada a objeto y el uso de la arquitectura Modelo Vista Controlador, hecho que a largo plazo facilita

cualquier labor de mantenimiento y diseño. Por otra parte, fue necesario el uso del lenguaje XML para la confección de las vistas (interfaz de usuario).

- Android es una excelente opción al momento de desarrollar software para dispositivos móviles, ya que proporciona acceso a una amplia gama de utilidades como el uso de mapas, lectura de códigos y la personalización de servicios todo mediante un número relativamente pequeño de instrucciones, lo cual explica el interés de muchas empresas que hoy en día apuestan por el uso de esta tecnología.
- La certeza de contar con el debido soporte y documentación necesarios para un buen desarrollo representa una de las principales ventajas de la plataforma. Sin lugar a dudas es una tecnología que posee un futuro prometedor y un catálogo de servicios que día a día va en aumento, gracias al desarrollo libre de licencias.
- Una de las desventajas detectadas de a la plataforma Android es el elevado costo de los teléfonos móviles en los cuales están integrados, lo cual se estima que a futuro disminuya tras hacerse masivo su uso.

Finalmente, debemos manifestar la total satisfacción por parte del equipo de desarrollo al cumplir con todas las funcionalidades establecidas en la planificación inicial del proyecto. Sin lugar a dudas la puesta en marcha real de esta iniciativa conllevaría a una mejora en la sincronización de los funcionarios de cualquier institución de la salud, eliminando los tiempos destinados a transcribir información y búsqueda de fichas médicas.

TRABAJOS FUTUROS

El desarrollo del proyecto fue realizado con el apoyo del centro de atención primaria Federico Puga Borne, pero no se realizó su puesta en marcha.

Como trabajo futuro sería importante instalar esta herramienta en un consultorio o clínica, haciendo uso de los registros de pacientes almacenados y de sus fichas médicas.

Utilizar otros servicios que ofrece Android, como reconocimiento de voz, enviar e-mail, mensajes de texto y multimedia,

Expandir la orientación de esta aplicación a otros rubros de trabajos en los cuales se requiera agendar o administrar datos, como por ejemplo, distribuidores de bebidas, comida etc.

Y por ultimo, hacer las pruebas de solicitud de 30 usuarios conectados al servidor simultáneamente, las cuales no pudieron ser llevados a cabo en esta ocasión por la falta equipos telefónicos disponibles.

BIBLIOGRAFÍA

1. Android. [En línea]. <<http://www.android-spa.com/>>[consulta: 8 de abril de 2010]
2. Android Maps API Key Signup.[En línea]<<http://code.google.com/intl/es-ES/android/maps-api-signup.html>> [Consulta: 10 de Julio 2010]
3. BURNETTE, Ed. *Hello Android*. 1a Edición, Pragmatig Bookshelft, 2008.
4. Breve introducción a la red de telefonía móvil . [En línea]. <<http://vidateleco.wordpress.com/2009/02/23/breve-introduccion-a-la-red-de-telefonía-movil/>> [Consulta: 08 de Abril de 2010]
5. FLING, Brian. *Mobile Design and Development*. 1a Edición, O'Reilly, 2009.
6. GAJARDO, LUIS. *Ecosistema Móvil*. [Diapositiva] <http://pva.face.ubiobio.cl/pva/file.php/508/diapositivas/02-EcosistemaMovil-new.pdf>, [2010].
7. GAJARDO, LUIS. *Introducción a la plataforma Android*. [Diapositiva] < <http://pva.face.ubiobio.cl/pva/file.php/508/diapositivas/04-Android-introduccion.pdf>> [2010].
8. LARMAN, Craig, *UML y Patrones*. Introducción al análisis y diseño orientado a objetos. 1ª Edición, Prentice Hall, 1999.
9. La penetración de la telefonía móvil en Chile. [En línea] <<http://www.laflecha.net/canales/moviles/noticias/la-penetracion-de-la-telefonía-movil-en-chile-alcanza-el-94-7>> [Consulta: 01 de Mayo de 2010]
10. PFLEEGER, Shari Lawrence. *Ingeniería de Software*. Teoría y Práctica. Prentice Hall, 2002.
11. Portillo. J y Carretero N. *Dispositivos portátiles y usabilidad*. [en línea] <<http://www.ceditec.etsi.upm.es>> [Consulta: 04 de Abril 2010]
12. PRESSMAN. *Ingeniería del Software: Un Enfoque Práctico*. 6a Edición. McGraw-Hill, 2002.
13. Red de Celdas. [En línea]< http://es.wikipedia.org/wiki/Red_de_celdas> [Consulta: 10 de abril 2010]
14. ROGERS, Rick et als. *Android Application Development*. 1a Edición, O'Reilly. 2009
15. SOMMERVILLE. *Ingeniería del Software*. 7a Edición, Addison-Wesley. Julio 2005
16. Telefonía móvil. [En línea] <<http://www2.udec.cl/~johperez/psi/tic/cuerpo.html>> [Consulta: 04 de Mayo de 2010]

ANEXOS

ANEXO A: CENTRO DE ATENCIÓN PRIMARIA FEDERICO PUGA BORNE

Corresponde al último avance en el área de la salud en la comuna de Chillán Viejo, es el primer centro de atención primaria que se desempeña como Centro de Salud Familiar (CESFAM). Abrió sus puertas a la comunidad en Julio del 2004, en él, se proporciona atención integral a nivel primario para el usuario y su familia, en la totalidad de su ciclo vital, lo que condiciona el mejoramiento de la salud y la mantención de hábitos de vida saludables.

En esta institución están inscritos 16047 pacientes, entrega atención primaria en base a un portafolio de servicios, el cual se programa anualmente con metas sanitarias establecidas según los compromisos de gestión y la realidad local.

La toma de decisiones se desarrolla dentro del marco que establece el Estatuto de Atención Primaria y la Ley Constitucional orgánica de municipalidades. Las decisiones técnicas son tomadas por el jefe del DESAMU y el equipo de profesionales. Las decisiones administrativas son tomadas en conjunto con el Alcalde, Administrador Municipal, Departamentos Municipales que tengan alguna responsabilidad en el tema de decisión y el Departamento de Salud Municipal. Para cumplir con su función el Consultorio, cuenta con una estructura de 2 pisos, el inferior es destinado al uso de consultorio médico, además del casino para el personal y baños públicos. En el segundo piso funcionan las oficinas administrativas. Es una construcción de tipo sismo-resistente, y de estilo moderno especialmente diseñado para el funcionamiento de un Centro de Salud Familiar.

Este hecho, sin duda, ha sido muy importante para la comunidad ya que ha facilitado el acceso de la población a mantener controles médicos permanentes (tanto de adultos, tercera edad como ancianos), curaciones, recetas, vacunaciones, etc.

Este centro de atención familiar cuenta con una excelente infraestructura tanto en calidad como modernidad, ya que trabaja con tecnología de punta tanto, a nivel administrativo (sistema computacional interconectado), como de equipamiento médico y odontológico.

Además otorga, atención de urgencia (SAPU), que permite aumentar la cobertura de pacientes en las zonas urbanas como rurales. También dispone de elementos audiovisuales de última generación, que favorece el proceso educativo de la comunidad, permitiendo la entrega de contenidos educativos de manera didáctica y amena, pudiendo dirigirse a diferentes grupos etéreos.

Anexo C: Ejemplo mini aplicación en Android

Primero se creará una vista en donde ingresaremos algunos datos, utilizando XML, que es el lenguaje definido para crear las interfaces.

En la imagen se puede observar como quedaría visualmente el código escrito.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    >
    <LinearLayout android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
        <TextView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Bienvenido a su Agenda médica"
            android:textColor="#FFFFFF"
            android:textSize="18px"
            />
    </LinearLayout>

    <LinearLayout android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
        <TextView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Ingrese su Rut : "
            android:textSize="12px"
            />
        <EditText
            android:id="@+id/rut"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:numeric="integer" />
        <TextView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Ej : 16.343.232-1"

            android:textSize="10px"
            />
    </LinearLayout>

    <LinearLayout android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
        <Button
            android:id="@+id/aceptar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Aceptar"
            android:layout_weight="1"/>
        <Button
            android:id="@+id/cancelar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Cancelar"
            />
    </LinearLayout>
</LinearLayout>
```



Figura 57, Aplicación Android

```

        android:layout_weight="1"
    />
</LinearLayout>
</LinearLayout>

```

Ahora veremos código java, que dependiendo del botón que se presione (Aceptar o cancelar) hace una determinada acción.

Aceptar : Escribe un saludo en el Editex.

Cancelar: Borra los datos del Editex.

```

package proyecto.chave;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class chave extends Activity {
    //Atributos
    private EditText rut;
    private Button cancelar;
    private Button aceptar;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        rut=(EditText) findViewById(R.id.rut);
        cancelar=(Button) findViewById(R.id.cancelar);
        aceptar=(Button) findViewById(R.id.aceptar);

        //Al presionar un botón cambia el editText
        aceptar.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                rut.setText("Ingreso datos");
            }
        });

        cancelar.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                setResult(RESULT_OK);
            }
        });
    }
}

```



Anexo D: Encuesta proceso de atención médica de pacientes a domicilio.

1. Instrucciones

- Todas las preguntas incluidas en la presente encuesta están relacionadas con el proceso de atención médica a domicilio llevado a cabo en el consultorio Dr. Federico Puga Borne.
- No es necesario que se preocupe por aspectos como la redacción, responda libremente según su experiencia y conocimiento.
- En el caso de no conocer la respuesta de alguna pregunta, sírvase dejar expresada tal situación, si conoce el rol del algún funcionario que pueda contribuir al ámbito de la pregunta tenga la amabilidad de indicar quién.

2. Preguntas

- 1) ¿Qué persona(s) puede(n) solicitar atención médica a domicilio?
- 2) ¿Que funcionario(s) es el encargado de asignar horas de atención a domicilio?
- 3) ¿Qué datos o documentos se requieren para solicitar atención a domicilio?
- 4) Si no existe la ficha médica de un paciente que requiere atención a domicilio ¿Cuál es el procedimiento?
- 5) ¿Cuales son las patologías que por lo general requieren de atención médica a domicilio?
- 6) ¿Cuántos pacientes a domicilio atiende aproximadamente el consultorio?
- 7) ¿Cada cuanto tiempo se realizan visitas a domicilio?
- 8) ¿Donde se realiza el registro de las horas de atención a domicilio?
- 9) ¿Qué factores se consideran para elegir el orden de las visitas?
- 10) ¿Quiénes conforman el equipo de visitas a domicilio?
- 11) ¿Cómo se coordina el equipo de visitas a domicilio?
- 12) ¿Qué documentación o datos se requieren al momento de visitar a un paciente a domicilio?
- 13) ¿Qué procedimiento es, a su parecer, engorroso o complejo de realizar al momento de visitar a un paciente?
- 14) ¿Existe la posibilidad de que un médico cancele una visita a domicilio?
- 15) ¿Cómo el personal médico se entera de las citas que debe realizar?

Anexo E: Prueba de usabilidad



UNIVERSIDAD DEL BÍO-BÍO

FACULTAD DE CIENCIAS EMPRESARIALES
 DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y
 TECNOLOGÍA DE LA INFORMACIÓN
 CAMPUS CHILLÁN

Prueba de Usabilidad
Agenda médica móvil sobre plataforma Android

En la siguiente tabla marca con una X el grado de facilidad que tuvo realizar la acción solicitada.

Acción a medir	Fácil	Normal	Difícil
1.- Identifíquese en el sistema, utilizando el Rut 164972135 y la clave 1234.			
2.- Vea las citas de día 19/07/2010.			
3.- Vea el perfil del paciente Raúl medina.			
4.- Vea el mapa con la ubicación de la dirección del paciente Raúl medina.			
5.- Agregue una nueva Cita médica al Rut 154972135.			
6.- Cree una nueva ficha médica con sus propios datos personales.			
7.- Cambie el estado de una visita médica.			
8.- Busque un paciente cuyo nombre sea Andrea.			
9.- Llame al paciente buscado anteriormente.			
10.- Ver detalle del historial de observaciones de Andrea.			

