



Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

**Aplicación móvil Android generadora de reportes
de reacción adversa a medicamentos, de apoyo
al médico, en la prescripción**

Autor: Daimary Hernández León

Tutores: MSc. Annia Arencibia Morales
Ing. Yoiler Joaquín Frometa Moreno

La Habana, 22 de junio del 2017

“Año 59 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro ser la autora de la presente tesis y reconozco a la Universidad de las Ciencias

Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Daimary Hernández León
(Autora)

MSc. Annia Arencibia Morales
(Tutor)

Ing. Yoiler Joaquín Frometa Moreno
(Tutor)

Datos de contacto

MSc. Annia Arencibia Morales (aarencibia@uci.cu): graduada de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el año 2007. Pertenece al Centro de Informática Médica (CESIM), actualmente posee la categoría docente de profesor Asistente. Máster en Informática Aplicada. Imparte Postgrado de Ingeniería de Software. Se desempeña como metodóloga de Práctica Profesional del CESIM.

Ing. Yoiler Joaquín Frometa Moreno (yfrometa@uci.cu): graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2007. Profesor de la Universidad de las Ciencias Informáticas, posee categoría docente de Profesor Instructor. Se desempeña como profesor de Almacenes de datos en la Facultad 2.

“Pienso que en vez de invertir tanto en el desarrollo de armas cada vez más sofisticadas, los que tienen los recursos para ello debieran promover las investigaciones médicas y poner al servicio de la humanidad los frutos de la ciencia, creando instrumentos de salud y de vida y no de muerte. No existiría entonces ni siquiera el pretexto para cobrar lo que se cobra por los medicamentos.”



Comandante Fidel Castro Ruz, en el acto de inauguración de las obras de ampliación del Hospital Clínico-quirúrgico “Joaquín Albarrán”, en ciudad de la Habana, el 5 de junio de 1989, “Año 31 de la Revolución”.

Agradecimiento

Agradezco a mi mamá María Onelia León Rojas y a mi papá Ibrahín Hernández Lugonez porque cuando no tenía a quien acudir, sabía que podía contar con ustedes, cuando todos los caminos se cerraban, su puerta era la única abierta y cuando todo se ponía difícil ahí estaban ustedes a mi lado diciéndome que todo saldría bien. Gracias mamá y papá por todo lo que hicieron y por todo lo que serían capaz de hacer si se los hubiera pedido. Sin ustedes no sería quien soy actualmente. Todo se los debo a ustedes.

A mi hermana Danay Hernández León por ser mi guía y porque desde que puedo recordar, has sido mi compañera en cada travesura, mi mejor amiga en todo momento, y mi paño de lágrimas. Te adoro, y sé que soy sumamente dichosa, por tenerte.

A mi hermano Alain Hernández Fernández pues aunque esté lejos siempre está conmigo y al tanto de mí.

A mi novio Pedro Pablo Álvarez Martínez que ha sido mi apoyo, mi sostén y mi vida. Gracias por soportarme y quererme sin condiciones, por estar siempre a mi lado en todos los momentos, pero sobretodo en los más difíciles, por hacerme la vida más fácil y alegre y porque sin tí no podría ser todo lo feliz que soy hoy, te amo.

A mis sobrinos por hacerme reír y ver la vida de otra perspectiva, porque sin ellos la vida y esta carrera no hubiesen sido lo mismo.

A Yoiler por aguantarme desde hace tanto tiempo, por apoyarme y por preocuparse por mí.

A mis tutores Annia y Yoiler que más que tutores son familia y han sabido apoyarme, guiarme y hasta regañarme cuando ha sido necesario.

A mi familia que me ha apoyado y creído siempre en mí.

A Fidel porque sin él y la Revolución no fuera hoy quien soy.

A todas las personas que de una forma u otra me han ayudado a estar hoy aquí.

Gracias a todos.

Dedicatoria

Le dedico esta tesis a mi mamá y a mi papá por estar conmigo en las buenas, en las malas y en las peores, por estar siempre dispuestos cuando necesitaba de ustedes, por abrirme los brazos cuando no tenía donde ir, por levantarme el ánimo cuando fracasaba, por confiar en mí cuando yo no confiaba ni en mí misma, por su fe sin límites y su amor infinito.

A mis hermanos por quererme y preocuparse por mí, por estar siempre cuando los he necesitado y por ser mis guías, los quiero.

A mi novio Pedro Pablo Álvarez Martínez por estar siempre a mi lado sin importar que tan difícil estuviera la situación, por la paciencia que ha tenido y por el amor que me ha brindado incondicionalmente.

A todos los amo.

Resumen

Las reacciones adversas a medicamentos son cualquier reacción nociva, no intencionada que aparece a dosis normalmente usadas en el ser humano. En Cuba se recoge la información de las reacciones a través del programa de notificaciones espontaneas, pero los médicos no tienen retroalimentación de esta información por lo que no la pueden tener en cuenta a la hora de prescribir el medicamento. El objetivo del siguiente trabajo es desarrollar una aplicación móvil Android, que permita a los médicos conocer a través de reportes de reacciones adversas los medicamentos que más reacciones causan en Cuba en un período de tiempo, sirviendo de apoyo a la prescripción.

Para lograrlo se desarrolló una aplicación móvil Android utilizando: como guía la metodología AUP-UCI, la herramienta Android Studio como IDE de desarrollo, Enterprise Architect para el modelado y Java como lenguaje de programación. Esta aplicación permite a los médicos consultar la información de los reportes de las reacciones adversas a medicamentos en el momento de prescripción, pudiendo así obtener conocimiento del comportamiento de las reacciones en un período de tiempo determinado y para un tipo de paciente específico. Lo que permite que el médico pueda prescribir un medicamento teniendo en cuenta la cantidad de reacciones que está causando, cumpliendo así con los objetivos del programa de farmacovigilancia cubano.

Luego de culminada la aplicación se le realizó pruebas de rendimiento, obteniéndose resultados satisfactorios, lo que permitió verificar que la aplicación funcionaba correctamente y que existe satisfacción por los futuros usuarios.

Palabras clave: Android, Farmacovigilancia, Reacción Adversa a Medicamentos, Reporte

Índice

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN	7
1.1 PRINCIPALES CONCEPTOS ASOCIADOS A LA FARMACOVIGILANCIA	7
1.2 SOLUCIONES EXISTENTES A NIVEL INTERNACIONAL Y NACIONAL	8
1.2.1 ADR reporting.....	9
1.2.2 Farmacovigilancia	9
1.2.3 Vademecum o medicamentos	9
1.2.4 FNM Cuba2016.....	9
1.3 ANÁLISIS DEL MERCADO DE DATOS DE SYNTA	10
1.4 SELECCIÓN DEL ENFOQUE DE LA METODOLOGÍA.....	13
1.5 AMBIENTE DE DESARROLLO	14
1.5.1 METODOLOGÍA DE DESARROLLO PROPUESTA	14
1.5.2 CONCEPTOS, HERRAMIENTAS, TÉCNICAS Y TECNOLOGÍA A UTILIZAR EN LA PROPUESTA DE SOLUCIÓN ...	15
1.6 CONCLUSIONES DEL CAPÍTULO.....	18
CAPÍTULO 2 PROPUESTA DE LA APLICACIÓN MÓVIL ANDROID RAMCUBA	19
2.1 PROPUESTA DE SOLUCIÓN	19
2.2 MODELO CONCEPTUAL	20
2.3 REQUISITOS DE SOFTWARE.....	21
<i>Requisitos funcionales</i>	22
<i>Requisitos no funcionales</i>	23
2.4. CASOS DE USO DEL SISTEMA.....	23
2.5 CONCLUSIONES DEL CAPÍTULO.....	35
CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA APLICACIÓN MÓVIL ANDROID RAMCUBA	36
1.1 PATRÓN ARQUITECTÓNICO	36
1.2 PATRONES DE DISEÑO	37
1.2.1 PATRONES GRASP.....	37
1.2.2 PATRONES DE DISEÑO GOF.....	39

1.3	MÉTRICAS PARA LA VERIFICACIÓN DEL DISEÑO DE CLASES.....	39
1.3.1	TAMAÑO OPERACIONAL DE CLASE (TOC).....	40
1.3.2	RELACIONES ENTRE CLASES (RC).....	42
1.4	MODELO DE DATOS	43
1.5	IMPLEMENTACIÓN.....	46
1.5.1	ESTÁNDAR DE CODIFICACIÓN	49
1.5.2	TRATAMIENTO DE ERRORES	51
1.6	PRUEBAS DE SOFTWARE.....	51
1.6.1	PRUEBAS DE CAJA NEGRA.....	52
1.6.2	PRUEBA DE RENDIMIENTO	52
1.7	APLICACIÓN DE LA TÉCNICA IADOV PARA MEDIR SATISFACCIÓN.....	52
1.8	CONCLUSIONES DEL CAPÍTULO.....	55
	CONCLUSIONES.....	56
	RECOMENDACIONES.....	57
	REFERENCIAS BIBLIOGRÁFICAS.....	58
	ANEXOS	62

Figura 1: Modelo de datos del mercado de datos del producto Synta	11
Figura 2 Resultado método de Boehm y Turner	14
Figura 3 Arquitectura del sistema operativo Android	16
Figura 4 Modelo Conceptual	21
Figura 5 Diagrama de casos de uso del sistema	24
Figura 6: Arquitectura modelo vista controlador	36
Figura 7 Resultado de la métrica TOC	41
Figura 8 Resultado de la métrica RC	43
Figura 9 Transformación dimensión especialidad	44
Figura 10 Modelo de datos de la propuesta de solución	45
Figura 11 Diagrama de clases del análisis del caso de uso generar cantidad por datos del paciente	46
Figura 12 Diagrama de comunicación del CU generar cantidad por datos del paciente	47
Figura 13: Diagrama de componentes del CU generar cantidad por datos del paciente	49
Figura 14: Diagrama de componentes	48
<i>Figura 15 Distribución de profesionales por años de experiencia.....</i>	<i>54</i>
Figura 16 Figura Distribución de profesionales por especialidad.....	55
Tabla 1. Comparación entre aplicaciones analizadas	10
Tabla 2 Dimensiones del mercado de datos de Synta	13
Tabla 3 Requisitos funcionales	22
Tabla 4 Requisitos no funcionales	23
Tabla 5 Métrica Tamaño operacional de las clases	41
Tabla 6 Umbrales utilizados en TOC	41
Tabla 7 Métrica Relaciones entre clases	42
Tabla 8 Umbrales utilizados en RC.....	42
Tabla 9 Resultado de la prueba de rendimiento	52
Tabla 10 Cuadro Lógico de IADOV	54

Introducción

Según la real academia de la lengua española, un medicamento es una “Sustancia que, administrada interior o exteriormente a un organismo animal, sirve para prevenir, curar o aliviar la enfermedad y corregir o reparar las secuelas de esta” (RAE- ASALE, 2014a). Pueden causar beneficios y reacciones adversas; los beneficios son los efectos favorables del uso de un medicamento en el organismo humano, como: calmar el dolor, bajar la presión arterial en los hipertensos, curar las infecciones, etc.(Errecalde, 2012)

La finalidad de los medicamentos es hacer que los pacientes se sientan mejor, curar o mejorar la salud, pero es importante saber que tienen reacciones adversas (AARP, 2015). Las reacciones adversas producidas por medicamentos son eventos no deseados o inesperados que ocurren cuando se administran estos para un tratamiento; los riesgos pueden ser leves como molestias del estómago, o puede ser más severos como el daño hepático, infarto de miocardio e inclusive provocar la muerte. (OMS, 2011a)

Un elevado porcentaje de la población ha sufrido algún tipo de reacción adversa a medicamentos (RAM), lo que representa un importante problema de salud pública debido a que pueden ser potencialmente mortales o provocar la prolongación de la hospitalización (OMS, 2011a). Las RAM según la Organización Mundial de la Salud (OMS) son: *«cualquier reacción nociva no intencionada que aparece a dosis normalmente usadas en el ser humano para profilaxis¹, diagnóstico o tratamiento o para modificar funciones fisiológicas»* (OMS, 2011b). Debido a esto la OMS crea el Programa Internacional de Farmacovigilancia para llevar el control de la posible relación entre el uso de un medicamento y sus efectos adversos. Actualmente cerca de cien países tienen sistemas nacionales de notificación de RAM, que envían informes anualmente al Centro Colaborador de la OMS para la Vigilancia Farmacéutica Internacional, radicado en Uppsala² (Suecia).(OMS, 2011b)

Este programa es llevado en Cuba por el Departamento de Farmacovigilancia del Ministerio de Salud Pública (MINSAP); encargado de la recolección, seguimiento, investigación y control de los posibles efectos adversos de los medicamentos (Morales et al., 2012) y tiene como objetivo general: *establecer un sistema que permita conocer la magnitud y consecuencias de las reacciones adversas al total de*

¹ Conjunto de medidas que se toman para proteger o preservar de las enfermedades.

² Uppsala: Es el Centro Colaborador de la Organización Mundial de la Salud para la Vigilancia Internacional de Medicamentos.

medicamentos consumidos por la población en el país. Los objetivos específicos de este programa son (López y Orta, 2015):

- ✓ Recoger, evaluar y registrar las sospechas de reacciones adversas a medicamentos notificadas.
- ✓ Medir la frecuencia de RAM según variables demográficas.
- ✓ Medir la frecuencia de RAM según fármacos y grupos farmacológicos.
- ✓ Medir la frecuencia de RAM graves.
- ✓ Identificar las RAM de baja frecuencia de aparición.
- ✓ Identificar las RAM evitables.
- ✓ Analizar la información existente con el fin de generar posibles señales.
- ✓ Identificar de forma precoz posibles riesgos asociados con el uso de medicamentos.
- ✓ Promover en los profesionales de la salud y los pacientes la necesidad e importancia de la farmacovigilancia, con el fin de estimular la notificación de RAM.
- ✓ Desarrollar un sistema de retroalimentación con los notificadores.

Para dar cumplimiento a los objetivos planteados por este programa, los farmacoepidemiólogos realizan informes de las reacciones adversas a medicamentos presentando: los medicamentos que más RAM están causando en un período determinado de tiempo para pacientes con una determinada edad, sexo, antecedente patológico y un motivo de prescripción, permitiendo realizar un análisis del comportamiento de las reacciones.(Orta y López, 2015)

La información de las RAM recogidas por el programa de notificación espontánea de Cuba es almacenada en una base de datos en formato Excel llamada FarmaVigiC; tras un análisis realizado a dicha base de datos con el objetivo de medir cuales medicamentos causaron más RAM en el período 2012 al 2015, se identificó que entre los diez primeros coinciden los siguientes:

- | | |
|------------------------|-----------------------------|
| ✓ Vacuna pentavalente | ✓ Amoxicilina |
| ✓ Captopril | ✓ Vacuna antimeningococcica |
| ✓ Penicilina rapilenta | ✓ Ciprofloxacina |
| ✓ Ibuprofeno | |

De los diez medicamentos que más RAM causaron en los años analizados coinciden siete, pero los profesionales sanitarios no conocen estos reportes y siguen prescribiéndolos, sin valorar cambiarlos por otro medicamento del mismo grupo farmacológico que no esté causando tantas reacciones, lo que puede incidir en la disminución de las RAM. Para homogenizar la información de FarmaVigiC y facilitar el análisis y confección de los informes que realizan los farmacoepidemiólogos, el MINSAP solicita realizar la

informatización de este departamento creándose el Sistema Integral para el Control Farmacológico (Synta); que permite realizar el control de los procesos de farmacoepidemiología integrados en un solo sistema (Morales et al., 2012). Synta cuenta con seis módulos: Tarjeta de Control, Receta Médica, Consumo de Medicamentos, Configuración, Subsistema Nomencladores y Reacciones Adversas a Medicamento, este último permite recoger la información de los efectos espontáneos desde la unidad de salud y cuenta con las siguientes características:

- ✓ Está compuesto por dos subsistemas: el Sistema inteligente para la predicción de RAM y un mercado de datos (Data Marts) para el análisis de la información (Morales, Pérez y Lorenzo, 2012).
- ✓ Este sistema soluciona los problemas de realizar un análisis detallado y preciso del comportamiento de las RAM en un momento determinado, ya sea por valores específicos o por la unión de varias características al mismo tiempo.
- ✓ Permite llevar el control de las RAM ocasionadas por los medicamentos consumidos por pacientes con un determinado Antecedente Patológico (APP).(Pérez y Lorenzo, 2012)
- ✓ Permite crear reportes por medicamentos, pacientes, grupo etario entre otras variables en un período de tiempo determinado.

En el año 2013 se decide desplegar el sistema Synta como parte de una prueba piloto en el municipio Guanabacoa de la provincia La Habana, por ser el municipio con mejor infraestructura tecnológica de La Habana. En una entrevista realizada al Ing. Ariel Sánchez Carrillón, jefe del departamento de informática en el municipio de Guanabacoa se identificó que: para un total de ciento veintidós unidades de salud, solo diez contaban con conectividad al Portal de la Red de Salud de Cuba (INFOMED)(donde se encuentra desplegado el sistema) y ciento doce consultorios médicos de familia (CMF) no pueden acceder al sistema porque no cuentan con conectividad, lo que deja un aproximado de ciento diecinueve médicos que no pueden tener acceso al sistema.

Para analizar cómo se comporta este problema en el resto del país. Esta encuesta se aplica ([ver anexo 1](#)) a los médicos del Hospital General Docente Danil Codorniu Pruna del municipio Placetas de Villa Clara y a los del Hospital Provincial Camilo Cienfuegos en Sancti Spíritus; con el objetivo de identificar los conocimientos del personal médico respecto a la existencia de Synta y de los medicamentos que más RAM causaron en el año 2015, además saber si creen necesario conocer las RAM ocurridas en Cuba. Los resultados obtenidos al procesar la encuesta fueron los siguientes:

- ✓ El 71.16% de los doctores desconocen la existencia de Synta.
- ✓ El 90.31% de estos médicos no conocen el medicamento que más RAM ocasionó en el año 2015.

- ✓ El 92.31% de los médicos cree necesario conocer las RAM ocurridas en Cuba en un período de tiempo.

Se concluye que Synta soluciona los problemas de homogenización y permite el análisis de la información, pero los médicos no conocen dicha información por lo que el problema persiste. Teniendo en cuenta esto los problemas existentes se describen a continuación:

- ✓ Al no conocer los médicos la información de los reportes respecto a las RAM generados por el MINSAP, ni tener un sistema en la consulta médica que les permita consultar en el momento de la prescripción qué medicamentos han causado más RAM, se indican medicamentos que pueden estar generando un número elevado de RAM. Esto incumple con la “estrategia de minimización” planteada por el programa de farmacovigilancia del MINSAP, teniendo como consecuencia que los reportes de RAM sigan aumentando cuando pueden minimizarse.
- ✓ Los pacientes tienen derecho a ser informados verazmente y de manera completa sobre los riesgos que para su salud competen los medicamentos que esté utilizando. Para evitar un mal mayor y solo de forma excepcional, podría justificarse la no comunicación total o parcial de la información. El médico al no conocer estos riesgos no podrá informar al paciente y este seguirá automedicándose, siendo esta una de las causas de aumento de los reportes de RAM.

Teniendo en cuenta lo anteriormente expuesto y la gran importancia que tiene para los médicos y farmacoepidemiólogos conocer los reportes de las RAM ocurridas en Cuba, se identificó el siguiente **problema a resolver**: los médicos desconocen los medicamentos que más reacciones adversas causan en Cuba, lo que trae consigo que no lo tengan en cuenta en la prescripción médica.

Identificándose como **objeto de estudio**: gestión de la información de las reacciones adversas ocurridas en Cuba. Enfocado en el **campo de acción**: generación de reportes de información de las reacciones adversas ocurridas en Cuba. Definiéndose como **objetivo general**: desarrollar una aplicación móvil Android, que permita a los médicos conocer a través de reportes los medicamentos que más reacciones adversas causan en Cuba, sirviendo de apoyo a la prescripción médica.

Para dar cumplimiento al objetivo general se plantean las siguientes **tareas de investigación**:

- ✓ Realizar el análisis del estado del arte de las soluciones existentes que presenten características semejantes a las de la investigación.
- ✓ Analizar el mercado de datos de Synta, con el objetivo de identificar posibles funcionalidades para la propuesta de solución.

- ✓ Seleccionar el enfoque y la metodología de desarrollo a utilizar para la realización de la propuesta de solución.
- ✓ Definir las herramientas, tecnologías y lenguajes para el desarrollo de la solución propuesta.
- ✓ Implementar la aplicación móvil Andorid, que permita a los médicos conocer a través de reportes los medicamentos que más reacciones adversas causan en Cuba en un período de tiempo.
- ✓ Validar la propuesta de solución para evaluar el correcto funcionamiento de la aplicación, mediante la prueba de caja negra, prueba de rendimiento y la satisfacción del usuario utilizando la técnica de IADOV.

Para el desarrollo de la investigación se utilizan los siguientes **métodos científicos**:

Métodos **teóricos** (León y González, 2011):

Histórico- lógico: se empleó para realizar un estudio analítico de la trayectoria histórica de la gestión de la información de las reacciones adversas ocurridas, con el objetivo de conocer la evolución, el estado y las tendencias actuales de la gestión de las RAM en diferentes entornos de Cuba y el mundo.

Analítico - Sintético: este método se empleó para descomponer el problema e identificar lo que trae consigo que los médicos no tengan en cuenta en la prescripción la información de los reportes.

Modelación: este método es utilizado para el modelado la solución propuesta.

Entre los métodos **empíricos** utilizados se encuentra:

A través de la **encuesta** se identificó con qué tipo de dispositivos cuentan los médicos e identificar que conocimiento tienen respecto a los medicamentos que más RAM ocasionaron en Cuba en el año 2015 y del sistema Synta ya implementado.

La **entrevista** se realizó con el objetivo de hacer un levantamiento de los resultados obtenidos de la prueba piloto del producto Synta en el municipio de Guanabacoa en el año 2013.

El contenido del presente trabajo está estructurado en tres capítulos distribuidos de la siguiente manera:

Capítulo 1: Fundamentación teórica de la investigación. Se describen los conceptos relacionados con el desarrollo de la investigación. Se lleva a cabo un análisis acerca de la gestión de la información de las reacciones adversas y herramientas informáticas a nivel internacional y nacional desarrolladas para el control farmacológico. Se especifica la metodología de desarrollo de software escogida y el ambiente de desarrollo.

Capítulo 2: Propuesta de la aplicación móvil Android RamCuba. Se describe la propuesta de solución cumpliendo con los requisitos del cliente, se generan los artefactos de la metodología AUP-UCI en el escenario seleccionado.

Capítulo 3: Implementación y Validación de la aplicación móvil Android RamCuba. En el presente capítulo se abordan los temas referentes al diseño, la implementación y puesta en práctica de la solución diseñada. Se muestran artefactos generados en la etapa de implementación, además, de los estándares de codificación utilizados en la construcción de la solución. Se detalla el trabajo acerca del tratamiento de errores y se plasma el diagrama de componentes. Se presenta además los resultados de los métodos de validación empleados.

Capítulo 1: Fundamentación teórica de la investigación

En este capítulo se muestran los conceptos relacionados con el desarrollo de la investigación. Se lleva a cabo un análisis acerca de la gestión de la información de las reacciones adversas y herramientas informáticas a nivel internacional y nacional para el control farmacológico. Se especifica la metodología de desarrollo de software escogida y el ambiente de desarrollo.

1.1 Principales conceptos asociados a la farmacovigilancia

A continuación, se desglosan los conceptos asociados a la farmacovigilancia para una mayor comprensión de la investigación.

Farmacovigilancia

“La farmacovigilancia es la ciencia y actividades relativas a la detección, evaluación, comprensión y prevención de los efectos adversos de los medicamentos o cualquier otro problema sugerido relacionados con ellos que, en un sentido estricto, se debe distinguir entre monitorización y farmacovigilancia. Conjunto de métodos que tienen por objeto la identificación, la evaluación cuantitativa del riesgo y la valoración cualitativa clínica de los efectos del uso agudo o crónico de los medicamentos en el conjunto de la población o en subgrupos específicos de ella.”(Orta y López, 2015)

El Sistema Cubano de Farmacovigilancia está integrado por una Unidad Coordinadora, ubicada en el Departamento de Farmacoepidemiología, dentro de la Dirección de Medicamentos y Tecnologías Médicas del Ministerio de Salud Pública (MINSAP), y un total de dieciséis responsables de farmacovigilancia provinciales, habiéndose implementado en todo el país un programa de farmacovigilancia, para todos los profesionales y técnicos de la salud, con un único método de validación e identificación de los riesgos a los medicamentos comercializados.(Orta y López, 2015)

El núcleo fundamental de trabajo del Sistema Cubano de Farmacovigilancia es la notificación de sospechas de reacciones adversas, a través del programa de notificación espontánea. Estas notificaciones son enviadas por los notificadores utilizando el modelo oficial del MINSAP 33-36-02 ([ver anexo 2](#)). La información recogida en este modelo es almacenada en una base de datos y se evalúa periódicamente por los especialistas en cada una de las instancias (provincial o nacional), con el fin de identificar de forma precoz posibles problemas de seguridad derivados del uso de los medicamentos (generación de señales de alertas).

Recogida de los reportes de RAM

Al proceso de la recogida de notificaciones espontáneas es llamado también en el MINSAP reportes de RAM. La recogida y gestión de la información de estos reportes se realiza de la siguiente manera:

- ✓ Los médicos o enfermeras registran en el modelo 33-36-02, la reacción y se almacenan en la Unidad de Salud (Orta y López, 2015).
- ✓ La base de datos FarmaVigiC (que se encuentra en formato Excel), es llenada por los farmacoepidemiólogos en los niveles de: municipio, provincia y nación (Orta y López, 2015).

Otros conceptos importantes asociados a la investigación.

Reporte: según la real academia de la lengua española un reporte es una noticia o un informe. (RAE-ASALE, 2014b)

Personal de salud: personal sanitario son «todas las personas que llevan a cabo tareas que tienen como principal finalidad promover la salud». (OMS, 2017)

Casos RAM: son los casos que se guardan en la base de datos FarmaVigiC de las RAM ocurridas en Cuba.

1.2 Soluciones existentes a nivel internacional y nacional

Para el análisis de las aplicaciones se tuvo en cuenta unas series de características que son relevantes para el desarrollo de la investigación, a continuación se detallan las que surgieron por necesidades de la investigación y las que se obtuvieron de la aplicación de la encuesta:

Por necesidades de la investigación que:

- ✓ Generación de gráficas, para que el médico pueda tener otra manera de ver los reportes.
- ✓ Realicen reportes de RAM, para que los médicos puedan conocer la información de los reportes de las reacciones en un período de tiempo determinado.
- ✓ Trabajen sin conexión, con el objetivo que se pueda usar desde consulta.
- ✓ Sea gratuita

Como resultado de la encuesta aplicada a los médicos se identificó el:

- ✓ Sistema operativo Android, teniendo en cuenta que el 83.72% de los encuestados tiene dispositivo móvil con esta tecnología.

Entre las aplicaciones analizadas no se encontró ninguna que se ajuste al campo de acción de esta investigación, por lo que se analizan aplicaciones similares dirigidas a las farmacovigilancia, entre ellas se encuentran:

1.2.1 ADR reporting

ADR reporting (SALUD, 2011) es una aplicación Android para teléfonos inteligentes, concebida por el Dr. Sachin Kuchya, Profesor Asociado en Farmacología en NSCB Medical College, Jabalpur (MP, India). Con la ayuda de la aplicación de informes ADR, médicos, enfermeras, farmacéuticos, pueden informar instantáneamente cualquier sospecha de reacciones adversas de toda India, esta aplicación requiere de conexión a internet para su funcionamiento, esta aplicación está diseñada para dispositivos con sistema operativo Android.

1.2.2 Farmacovigilancia

Farmacovigilancia (UNAM, 2016) es un cuestionario de informes de sospechas de reacciones adversas de medicamentos. Permite realizar reportes de sospechas de reacción adversa a medicamentos e incluye datos sobre el medicamento sospechoso, historia clínica del paciente. Esta aplicación requiere de conexión a internet para su funcionamiento, esta aplicación está diseñada para dispositivos con sistema operativo IOS.

1.2.3 Vademecum o medicamentos

Permite consultar de forma gratuita más de 30.000 fármacos y 8.000 principios activos presentes en más de 30 países. No necesita acceso a Internet. Permite buscar por principio activo o medicamentos de forma simple. Brinda alertas de farmacovigilancia, lactancia, embarazo, insuficiencia renal y hepática, fotosensibilidad y efectos en la conducción, esta aplicación está diseñada para dispositivos con sistema operativo Android. (Martín, 2015).

1.2.4 FNM Cuba2016

Esta aplicación surge como complemento del trabajo realizado por los especialistas del Comité Editorial del Formulario Nacional de Medicamentos de Cuba, cuyo contenido está disponible en formato impreso y digital en línea, como fuente de información de la Biblioteca Virtual en Salud de Cuba perteneciente al Centro Nacional de Información de Ciencias Médicas-Infomed. Con el uso de las nuevas tecnologías, se extiende a los dispositivos con sistema operativo Android y puede utilizarse sin conexión. Consta de una base de datos con 750 fármacos empleados en el Sistema Nacional de Salud de Cuba y 500 principios activos. La información está organizada por orden alfabético, presentaciones, categorías farmacológicas y

riesgos para grupos especiales, entre otras modalidades. A través de su opción de búsqueda se pueden localizar los medicamentos y plantas medicinales. Incluye además, un sistema de escaneo digital para la identificación de gran número de fármacos por su codificación de barras y brinda información necesaria como las categorías de riesgo para lactancia y embarazo (FNM, 2017).

Comparación de las aplicaciones descritas

Tabla 1. Comparación entre aplicaciones analizadas
Fuente: elaboración propia

Aplicaciones	1	2	3	4	5
ADR reporting	x	x		X	
Farmacovigilancia		x			
Vademecum o medicamentos	x		x		
FNM Cuba2016	x	x	x		

Leyenda: 1. Sistema operativo Android, 2. Gratuitas, 3. Trabajo sin conexión, 4. Generan gráficas, 5. Realicen reportes de RAM

De las aplicaciones analizadas el FNM Cuba2016 es el más semejante a la investigación teniendo en cuenta que cumple con tres de cinco características, pero no puede ser utilizado teniendo en cuenta que no cumple con la característica fundamental de esta investigación, que es la generación de reportes de casos de RAM.

Otra de las necesidades de la investigación es el origen de los datos a utilizar y para esto se analiza el Mercado de datos del módulo Reacción Adversa a Medicamento (MRAM) del producto Synta.

1.3 Análisis del mercado de datos de Synta

En el análisis realizado se obtuvo que el mismo cuenta con 22 dimensiones relacionadas con el hecho_registro_ram que es donde se almacena cada aparición de RAM en el modelo de datos relacionado se muestra en la Figura 1: Modelo de datos del mercado de datos del producto Synta.

para evitar que afecte demasiado a la población (este análisis actualmente se hace, pero no se pueden tomar medidas a tiempo).

- ✓ Es necesario llevar el control de las RAM ocasionadas por los medicamentos consumidos por pacientes con un Antecedente Patológico (APP) determinado hipertensión, asma, diabetes, entre otros); identificando si algunas de estas RAM no son frecuentes, para realizar un análisis de inmediato.
- ✓ Las embarazadas, puérperas o lactantes, deben tener un seguimiento especial respecto a las RAM ocurridas durante estas etapas (algunas pueden ocurrir por el consumo inadecuado de un medicamento, la inadecuada vacunación en una etapa de gestación o simplemente porque un medicamento está ocasionando RAM no previstas). Estos pacientes presentan condiciones únicas, sobre las que se deben tomar medidas de forma rápida, que permitan salvaguardar tanto la vida de la madre, como la del niño.
- ✓ Total de RAM por sexo, causalidad.
- ✓ Cantidad de pacientes afectados por una RAM por raza y especialidad.

Este mercado de datos está desarrollado bajo la metodología Hefesto y la base de datos de la que obtiene información en postgres. Esta base de datos es la del producto Synta específicamente la información referente a las RAM. Esta información es tomada a partir de los registros del Módulo de Reacción Adversa a Medicamentos (MRAM); debido a que un conjunto de esta información es nomencrada, es necesario obtenerla del subsistema Nomenclador, este sistema maneja toda la información nomencrada de Synta entre la que se encuentra: los medicamentos, las RAM, grupo etareos entre otras. Además se necesita la clasificación VEN³ de los medicamentos que se encuentra registrada por el Módulo Consumo de Medicamentos).

³ VEN: clasificación de los medicamentos, teniendo en cuenta si son: vitales, no vitales, esenciales o especiales.

Toda la información almacenada en este mercado de datos se encuentra en distintas dimensiones que se muestran a continuación:

Tabla 2 Dimensiones del mercado de datos de Synta
Fuente: elaboración propia

Dimensiones	
dim_raza	dim_decenlace
dim_severidad	dim_ram
dimsexo	dim_especialidad
dim_tiempo	dim_medicamento
dim_app	dim_causalidad
dim_ven	dim_paciente
dim_grupo_edad	dim_nivel_de_atención
dim_localizacin	dim_datos_registrados
dim_frecuencia_ram	dim_secuencia_temporal
dim_calidad_e1	dim_calidad_e2
dim_calidad_e3	hecho_registro_ram

Para el desarrollo de la aplicación es necesario seleccionar enfoque de metodología, metodología de desarrollo, herramientas y arquitectura.

1.4 Selección del enfoque de la metodología

Para la selección del enfoque de la metodología se utilizó el método de Boehm y Turner que permite caracterizar el proyecto de software a partir de 5 criterios y estimar cuan ágil o prescriptivo debía ser el enfoque a utilizar. Para la selección del valor que se ubicará en cada eje de la estrella se debe tener en cuenta el comportamiento de estos criterios en el proyecto. En lo sucesivo se describe cada uno: (Kuhrmann, Münch y Rausch, 2016)

- ✓ Tamaño: este criterio se utiliza para representar el número de personas involucradas en el proyecto. Teniendo esto en cuenta, se le dio valor tres porque el desarrollo de la aplicación es de una sola persona con poca experiencia.
- ✓ Criticidad: se utiliza para evaluar la naturaleza del daño ocasionado por defectos que no hayan sido detectados al producto. Su evaluación puede ser cualitativa. Teniendo esto en cuenta la evaluación fue pequeña puesto que no es un sistema crítico.
- ✓ Dinamismo: representa la rapidez con la que pueden estar cambiando los requerimientos del proyecto. La aplicación a desarrollar está basada en una investigación anterior por lo que los requisitos no deben variar mucho, por lo que se le da valor cinco.

- ✓ Personal: el por ciento del personal con: capacidad, experiencia y posibilidades para enfrentar tareas en la investigación se le otorga valor cero debido a que es una sola persona con poca experiencia.
- ✓ Cultura: las organizaciones y las personas que relaciona el proyecto pueden depender de la confianza o de la relación contractual. Esto refleja el nivel de ceremonia necesario y aceptado: documentación, control, formalismo en las comunicaciones, por lo que se le da valor 90 como la aplicación que se va a desarrollar es una aplicación móvil no necesita de permisos administrativos para su integración en la comunidad.

Al aplicar el método los cinco valores de la estrella demuestran el enfoque, mientras más cerca esté del centro indica que la metodología a utilizar tiene que ser ágil. En este caso como se muestra en la Figura 2 Resultado de Boehm y Turner, los puntos de la estrella formada están pegadas al centro, indicando que en dicha investigación se debe usar una metodología ágil.

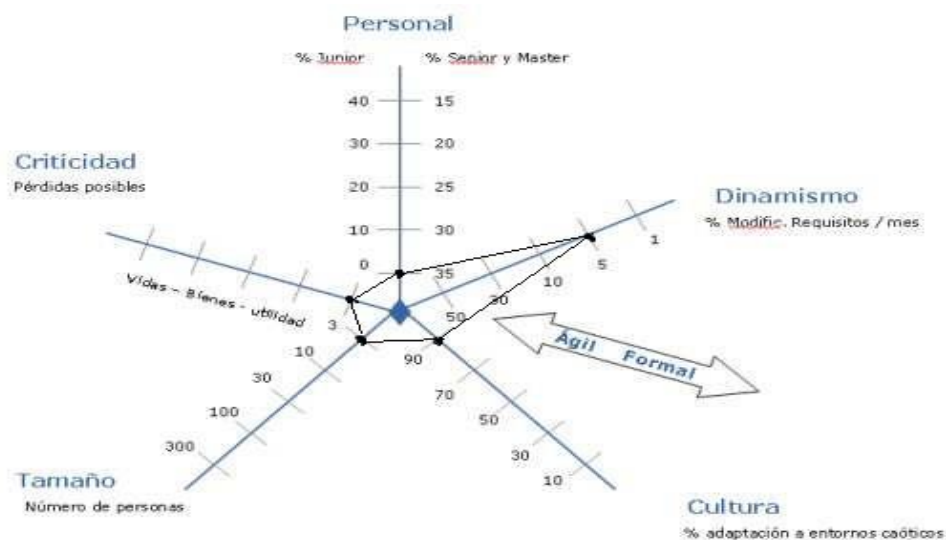


Figura 2 Resultado método de Boehm y Turner
Fuente: elaboración propia

1.5 Ambiente de desarrollo

A continuación se describe el ambiente de desarrollo teniendo en cuenta la metodología, lenguaje, herramientas y tecnologías a utilizar.

1.5.1 Metodología de desarrollo propuesta

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP), este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos

que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo:

- ✓ Desarrollo Dirigido por Pruebas (test driven development - TDD en inglés)
- ✓ Modelado ágil
- ✓ Gestión de Cambios ágil
- ✓ Refactorización de Base de Datos para mejorar la productividad.

La metodología propuesta para el desarrollo de la aplicación es **AUP-UCI** en el **segundo escenario** que plantea “Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información ” (Sánchez, 2015). Se escoge esta metodología porque presenta un enfoque ágil y se ajusta a las necesidades de la investigación.

1.5.2 Conceptos, herramientas, técnicas y tecnología a utilizar en la propuesta de solución

Para el desarrollo de la aplicación se utilizaron un conjunto de conceptos, técnicas, tecnologías y herramientas, la cuales se describen a continuación.

Aplicaciones móviles

Una aplicación móvil o ‘app’ es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Por lo general se encuentran disponibles a través de plataformas de distribución, operadas por las compañías propietarias de los sistemas operativos móviles como Android, iOS, BlackBerry OS, Windows Phone, entre otros (Gironés, 2012). Se puede definir una aplicación móvil como “un elemento ejecutable que primeramente se descarga e instala en un dispositivo y posteriormente se utiliza para el fin que haya sido creado”. (Gironés, 2012)

Android

Las aplicaciones necesitan un sistema operativo móvil para ejecutarse y transmitir instrucciones al hardware. Entre los sistemas operativos que se usan hoy día uno de los que sobresale es Android, el mismo está desarrollado con un núcleo de Linux. (Soriano, 2011)

Características de Android

Las características que diferencian a Android de los demás sistemas operativos: (Gironés, 2012)

- ✓ Es una plataforma abierta. Es de desarrollo libre, basado en Linux.
- ✓ Se puede adaptar a cualquier tipo de *hardware*. Actualmente el uso más extendido de Android se centra en móviles y *tablets*, pero el sistema operativo Android puede ser usado en relojes o electrodomésticos.
- ✓ Una aplicación diseñada para una determinada versión de Android funciona en todos los dispositivos que se ejecutan en esa versión.
- ✓ Optimizado para baja potencia y poca memoria. Debido al uso de una máquina virtual Dalvik.

La arquitectura del sistema operativo Android se define como se describe en la siguiente figura 2: Arquitectura del sistema operativo Android

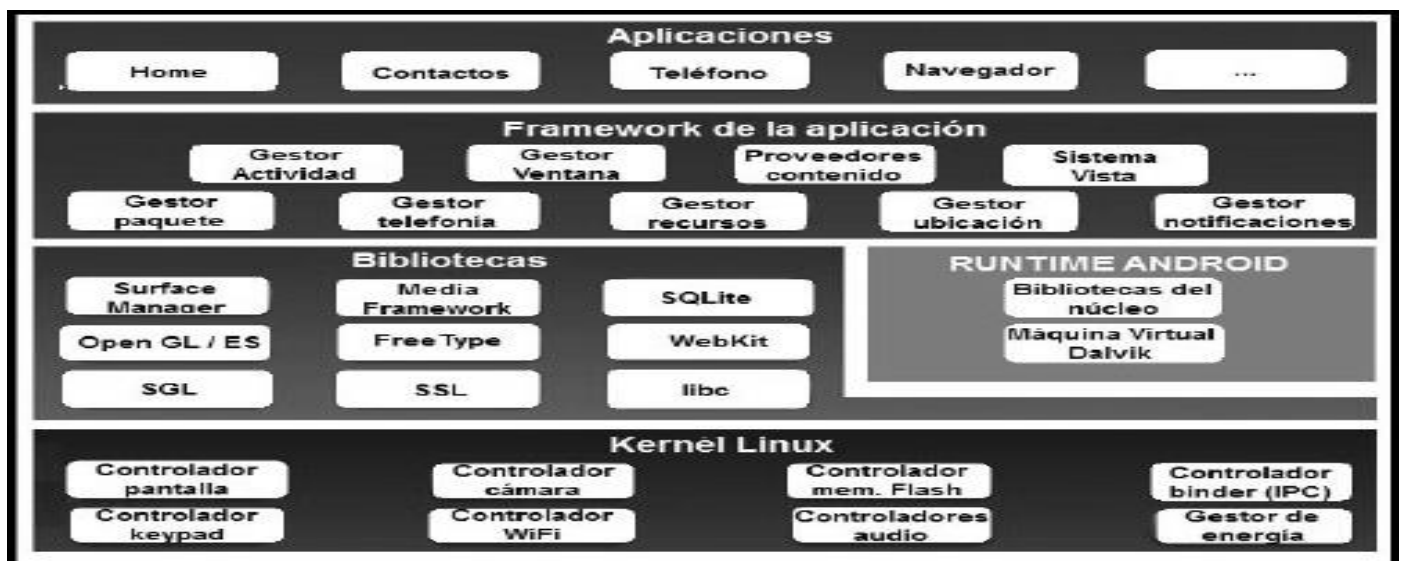


Figura 3 Arquitectura del sistema operativo Android
Fuente: (Pérochon, 2014)

Otros conceptos asociados a la investigación que son de gran importancia para una mayor comprensión son:

Lenguaje de modelado UML v2.1

UML es un lenguaje gráfico para especificar, construir y documentar los artefactos que modelan un sistema. UML fue diseñado para ser un lenguaje de modelado de propósito general, por lo que puede utilizarse para especificar la mayoría de los sistemas basados en objetos o en componentes, y para modelar aplicaciones de muy diversos dominios de aplicación (telecomunicaciones, comercio, sanidad, etc.) y plataformas de objetos distribuidos. (Palomo y Gil, 2014a)

Enterprise Architect 7.1

Enterprise Architect 7.1 (EA) es una herramienta de análisis y diseño UML comprensivo, cubriendo el desarrollo de software desde la obtención de los requisitos, diseño del modelo, pruebas, cambio de control y mantenimiento para la implementación, con completa trazabilidad. EA combina el poder de la especificación UML 2.1 con alta *performance*, una interfaz intuitiva, para brindar un modelado avanzado a todo el equipo de desarrollo. Es una herramienta visual con multi-usuario con un gran establecimiento de características, ayudando a los analistas, *testers*, administradores de proyectos, personal del control de calidad y desarrolladores alrededor del mundo a construir y documentar, software sostenible. (Ortega, Villa y Andrea, 2012).

Lenguajes de programación Java

Java es un lenguaje de programación orientado a objetos, el cual agrupa en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. Diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución (Sala, 2013). Se determinó utilizar Java 7 para la implementación de la Aplicación Móvil teniendo en cuenta que es un lenguaje orientado a objetos de gran potencialidad, robustez, seguro y multiplataforma Además, porque es el lenguaje nativo en el desarrollo de aplicaciones Android.

Android Studio como entorno de desarrollo integrado (IDE)

Un IDE es un editor y un compilador, junto con muchas características adicionales destinadas a brindar comodidad y velocidad al programador (Gironés, 2012). Android Studio es un IDE basado en IntelliJ IDEA de la compañía JetBrains, proporciona varias mejoras con respecto al plugin Android Developer Tools para Eclipse. Utiliza una licencia de software libre Apache 2.0, está programado en Java y es multiplataforma.(Smyth, 2017)

Software Development Kit (SDK)

SDK en español Kit de Desarrollo de Software (Ayala, 2015) es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto. El SDK es una API⁴ creada para permitir el uso de cierto lenguaje de programación, o puede,

⁴ La interfaz de programación de aplicaciones, abreviada como **API** del inglés: Application Programming Interface, es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

también, incluir hardware sofisticado para comunicarse con un determinado sistema embebido⁵. Las herramientas más comunes incluyen soporte para la detección de errores de programación IDE y otras utilidades.

1.6 Conclusiones del capítulo

Después de la realización del capítulo uno se concluye lo siguiente:

- ✓ A través del análisis de soluciones existentes, se determinó que ninguna podría ser utilizada para la solución del problema, por lo que se hace necesaria la implementación de una nueva aplicación que cumpla con las necesidades de la investigación, pero sirven de base para la solución de problema.
- ✓ A través de la aplicación del método de Bohem y Turner, se pudo determinar que el enfoque a utilizar es ágil.

⁵ Un **sistema embebido** (anglicismo de embedded) o empotrado (integrado, incrustado) es un **sistema** de computación diseñado para realizar una o algunas pocas funciones dedicadas, frecuentemente en un **sistema** de computación en tiempo real.

Capítulo 2 Propuesta de la aplicación móvil Android RamCuba

En el presente capítulo se expone la propuesta de solución. Generando los artefactos requeridos por la metodología AUP-UCI en su segundo escenario.

2.1 Propuesta de solución

La solución propuesta consiste en el desarrollo de una aplicación móvil Android que permita realizar reportes de reacciones adversas a medicamentos ocurridas en Cuba, estos reportes permitirán a los médicos tener un referente que les sirva de apoyo en la prescripción teniendo en cuenta los medicamentos que más RAM están causando. La aplicación permitirá visualizar los reportes de forma gráfica y nominal, dichos reportes se podrán realizar por distintos indicadores como son:

- ✓ RAM por años: permitiendo al personal de salud conocer el comportamiento de una RAM en un período de tiempo.
- ✓ RAM por medicamentos: permite conocer que medicamento está causando más reacciones y que reacción fue causada por un medicamento en un período de tiempo.
- ✓ RAM por datos del paciente: permite conocer la cantidad de RAM ocurridas en un período de tiempo por los indicadores asociados a los pacientes.
- ✓ RAM por tipo de paciente: permite conocer la cantidad de embarazadas, puérperas y lactantes que sufrieron RAM en un determinado período de tiempo.
- ✓ RAM por datos generales: permite conocer la cantidad de RAM ocurridas en un período de tiempo por los indicadores asociados a la RAM como especialidad, nivel de atención, desenlace, frecuencia entre otros.
- ✓ RAM por datos registrados : permite conocer la cantidad de RAM ocurridas en un período de tiempo referentes a indicadores tales como si sufrió reposo por más de tres días, baja laboral o escolar, peligro de vida, ingreso, atención a urgencia, entre otros.
- ✓ Porcentaje de pacientes por RAM: permite conocer qué por ciento representa un grupo poblacional afectado por una RAM en un período de tiempo, los grupos pueden elaborarse por indicadores como sexo, raza y grupo etario.

La aplicación contará con una pantalla principal que le permitirá al usuario seleccionar el tipo de reporte que desea generar. Luego de seleccionado el reporte se mostrará una interfaz donde se podrán

seleccionar los indicadores necesarios para generar el reporte seleccionado. Los datos entrados se enviarán a un visualizador que será el encargado de mostrar los resultados. La interfaz de indicadores contará con campos de selección y de autocompletado que permitirá que el usuario cometa la menor cantidad de errores posibles a la hora de introducir los datos, además, los intervalos de fechas estarán validados para evitar irregularidades. En cada pantalla existirá un vínculo a la interfaz principal de la aplicación y un botón que brindará la opción de limpiar los campos llenos.

Para realizar los reportes se utilizará la base de datos del mercado de datos integrado al sistema Synta, con el objetivo de manejar la misma información que el sistema. Esta base de datos se encuentra en postgres y debe ser migrada a sqlite que es el gestor de base de datos embebido en Android; para la migración de la estructura se utilizará el software OpenDbCopy⁶ y para los datos se usará la herramienta Pentaho Data Integration.⁷ Se decide proteger la información, por lo que la base de datos (BD) se cifrará en Base64⁸ y el usuario solo tendrá acceso a la base de datos a través de los resultados de los reportes.

2.2 Modelo conceptual

Modelo conceptual, modelo de objetos del dominio y modelo de los objetos de análisis son distintos nombres que suelen usarse indistintamente para referirse a este modelo (Amo, Normand y Pérez, 2013). Un modelo conceptual tiene como objetivo *“identificar y explicar los conceptos significativos en un dominio de problema, identificando los atributos y las asociaciones existentes entre ellos”* (Palomo y Gil, 2014b). En este modelo se definen cuáles son y cómo se relacionan los conceptos relevantes en la descripción del problema, en este caso se describen los conceptos relacionados con el negocio del sistema.

A continuación, se describen los conceptos fundamentales que componen el modelo conceptual del negocio:

- ✓ Personal de salud: usuarios que generan los reportes de las RAM enfermera, médico y farmacoepidemiólogo.

⁶ OpenDbCopy: es una herramienta con licencia OpenSource que permite replicar la estructura de una base de datos en distintos lenguajes.

⁷ Pentaho Data Integration: es una herramienta de Business Intelligence desarrollada bajo la filosofía del software libre para la gestión y toma de decisiones empresariales.

⁸ Base64 es un grupo de esquemas de codificación de binario a texto que representa los datos binarios mediante una cadena ASCII, traduciéndolos en una representación radix-64.

- ✓ Casos de RAM: son el conjunto de valores que se recogen de una persona que haya sufrido una RAM.
- ✓ Reporte de RAM: son el conjunto de valores que se recogen de una RAM en los reportes espontáneos realizados por el personal de salud.

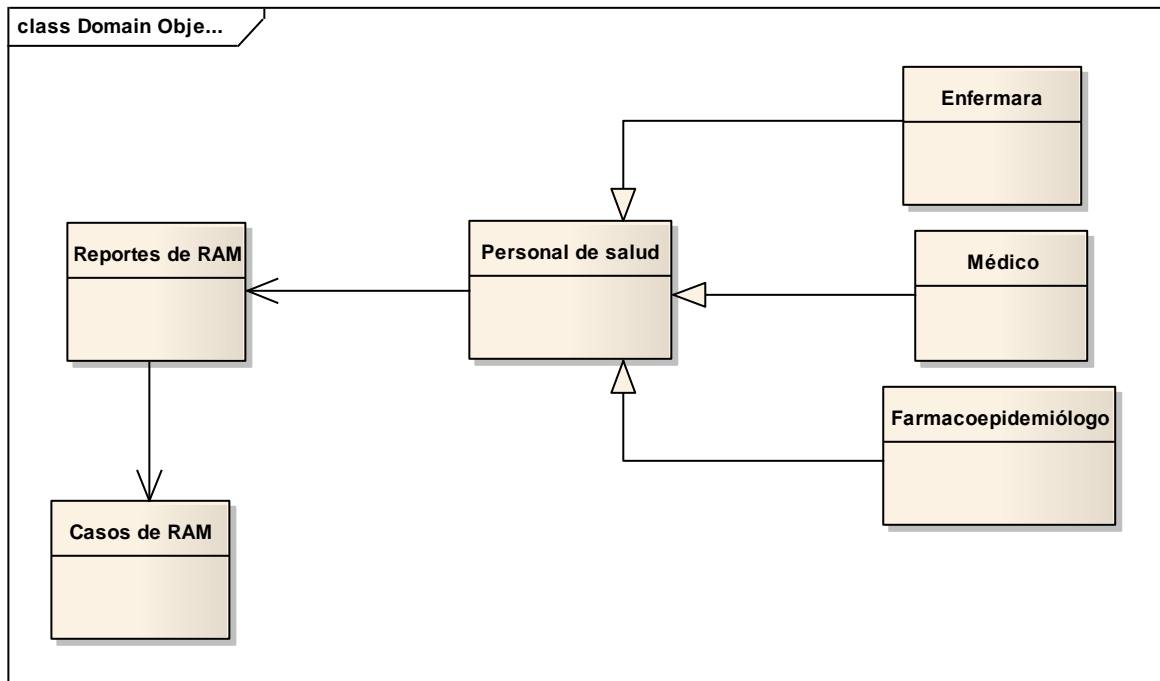


Figura 4 Modelo Conceptual
Fuente: elaboración propia

2.3 Requisitos de software

Somerville define: “Un requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste. En el otro extremo, es una definición detallada y formal de una función del sistema” (Sommerville, 2011a). Los requisitos de software se clasifican en funcionales y no funcionales. Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir y los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. (Pressman, 2010)

Requisitos funcionales

Tabla 3 *Requisitos funcionales*
Fuente: elaboración propia

Número	Requisitos funcionales
RF1	Generar reporte de RAM en un período de tiempo
RF2	Generar reporte de RAM por medicamentos
RF3	Generar reporte de la cantidad de pacientes, afectados por una RAM, por sexo y raza, en un período de tiempo determinado
RF4	Generar reporte de la cantidad de pacientes, afectados por una RAM, por grupo etario y edad en un período de tiempo determinado
RF5	Generar reporte de la cantidad de pacientes, afectados por una RAM, por antecedentes patológicos en un período de tiempo determinado
RF6	Generar reporte de la cantidad de pacientes, afectados por una RAM, por localización en un período de tiempo determinado
RF7	Generar reporte de la cantidad de pacientes, afectados por una RAM, por medicamentos en un período de tiempo determinado
RF8	Generar reporte del porcentaje de pacientes, afectados por una RAM, por medicamentos en un período de tiempo determinado
RF9	Generar reporte del porcentaje de pacientes, afectados por una RAM, por sexo, raza, grupo etario y edad en un período de tiempo determinado
RF10	Generar reporte del total de registros RAM, por niveles de atención, especialidad y localización, en un período de tiempo determinado
RF11	Generar reporte del total de registro RAM, por frecuencia, desenlace, causalidad y severidad en un período de tiempo determinado
RF12	Generar reporte del total de registro RAM, por medicamentos en un período de tiempo determinado
RF13	Generar reporte del total de registro RAM, por secuencia temporal y datos registrados en un período de tiempo determinado
RF14	Generar reporte de la cantidad de medicamentos, por registro RAM, en un período de tiempo determinado
RF15	Generar reporte de la cantidad de medicamentos que han ocasionado una RAM a un

	paciente determinado en un período de tiempo determinado
RF16	Generar reporte de la cantidad de medicamentos que han ocasionado RAM, según su clasificación VEN en un período de tiempo determinado
RF17	Graficar los reportes realizados
RF18	Limpiar los campos

Requisitos no funcionales

Tabla 4 Requisitos no funcionales

Fuente: elaboración propia

Número	Requisito no funcional de usabilidad
RNF1	Las interfaces del sistema deben ser entendibles, sencillas y amigables para que se pueda acceder a las funcionalidades de una forma rápida y cómoda.
Requisito no funcional de software	
RNF2	Para un correcto funcionamiento en el dispositivo se recomienda tener Android 2.2 o superior.
Requisito no funcional de hardware	
RNF3	Para un correcto funcionamiento en el dispositivo cliente se recomienda tener 256 de RAM o superior
RNF4	Para un correcto funcionamiento en el dispositivo cliente se recomienda tener como mínimo 15 MB de almacenamiento
Requisito no funcional de Seguridad	
RNF4	La base de datos se encontrara cifrada en Base64 para que la información solo pueda ser accedida a través de los reportes

2.4. Casos de uso del sistema

Los diagramas de casos de uso del sistema (CUS), describen los intercambios entre el sistema que se está describiendo y las personas o sistemas externos que interactúan con el primero, por lo tanto son muy útiles para describir funcionalidades a varios tipos de usuarios y con muchas interfaces (Burgués, 2016).

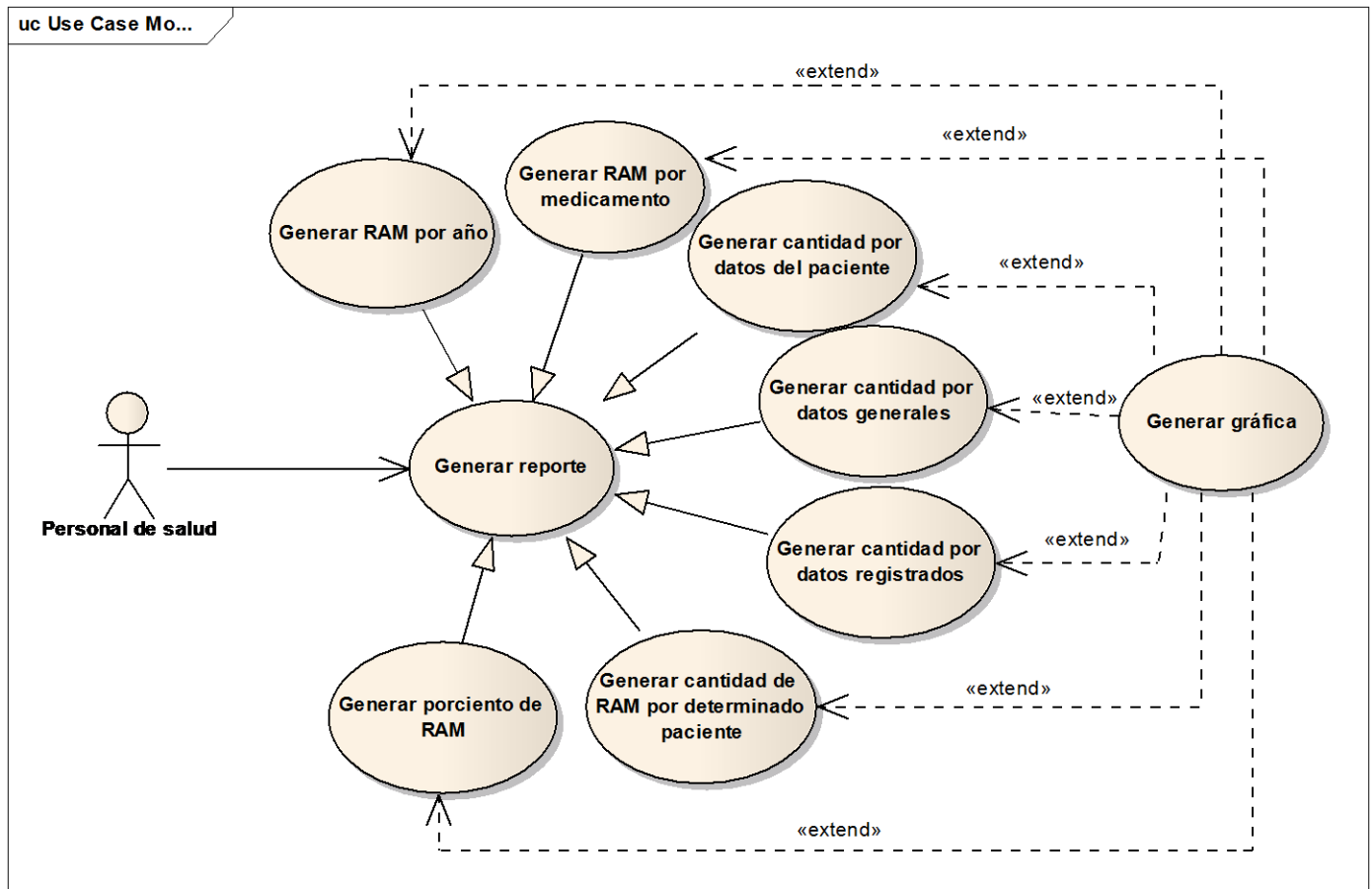


Figura 5 Diagrama de casos de uso del sistema

Fuente: elaboración propia

Para una mayor comprensión de los casos de uso se realiza la descripción textual de casos de uso que se muestra a continuación.

Descripción textual de los casos de uso del sistema

Una descripción textual expresa una secuencia de transacciones que son desarrolladas por un grupo de trabajadores, usuarios o una solución informática respectivamente, en respuesta a un evento que inicia uno de estos actores sobre el proceso de negocio o la propia solución informática. (Burgués, 2016)

Los casos de uso evitan típicamente el lenguaje técnico, prefiriendo la lengua del usuario final o del experto del campo del saber al que se va a aplicar. Los casos del uso son a menudo elaborados en colaboración por los analistas de requerimientos y los clientes. (SOLÍS y TINOCO, 2015)

Cada caso de uso se centra en describir cómo alcanzar una única meta o tarea. Desde una perspectiva tradicional de la ingeniería de software, un caso de uso describe una característica del sistema. Para la mayoría de proyectos de software, esto significa que quizás a veces es necesario especificar decenas o centenares de casos de uso para definir completamente el nuevo sistema. El grado de la formalidad de un proyecto particular del software y de la etapa del proyecto influenciará el nivel del detalle requerido en cada caso de uso.(Martínez, 2017)

CU1: Generar reporte.

Objetivo	Generar reportes sobre reacciones adversas a medicamentos	
Usuario	Personal de salud	
Resumen	El caso de uso inicia cuando el usuario desea generar un reporte y selecciona el tipo de reporte que desea generar.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	Tener información sobre el reporte seleccionado.	
Postcondiciones	Reporte generado	
Flujo de eventos		
Flujo básico		
Personal de salud	Sistema	
1 Inicia la aplicación RAM Cuba	Muestra un menú con las siguientes opciones: <ul style="list-style-type: none"> ▪ RAM por año ▪ RAM por medicamento ▪ RAM por datos del paciente ▪ RAM por datos generales ▪ Cantidad de RAM por datos registrados ▪ Cantidad de RAM por paciente ▪ Porcentaje de pacientes por RAM 	
2 Selecciona la opción de reporte deseada.	Si selecciona: <ul style="list-style-type: none"> ▪ RAM por año se ejecuta el CUS2: Generar RAM por año ▪ RAM por medicamento se ejecuta el CUS3 	

	<p>Generar RAM por medicamento</p> <ul style="list-style-type: none"> ▪ RAM por datos del paciente se ejecuta el CUS4 Generar cantidad por datos del paciente ▪ RAM por datos generales ejecuta el CUS5 Generar cantidad por datos generales ▪ Cantidad de RAM por datos registrados se ejecuta el CUS6 Generar Cantidad por datos registrados ▪ Cantidad de RAM por paciente se ejecuta CUS7 Generar Cantidad de RAM por un determinado paciente ▪ Porciento de pacientes por RAM se ejecuta CUS8 Generar Porciento de RAM
<p>3 Presiona el botón Home para ir a la interfaz menú.</p>	<p>Muestra la interfaz menú</p>
<p>Prototipo</p>	

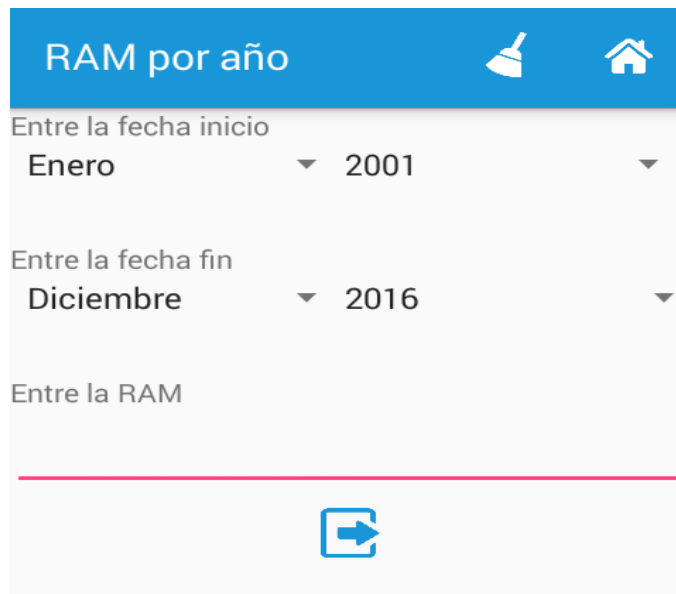


CU2: Generar RAM por año

Objetivo	Generar la cantidad de reacciones adversas a medicamentos en un período de tiempo determinado.	
Usuario	Personal de salud	
Resumen	El caso de uso inicia cuando el usuario desea generar un reporte sobre las RAM ocurridas en un período de tiempo determinado.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	Tener información sobre las RAM de período de tiempo.	
Postcondiciones	Reporte generado	
Flujo de eventos		
Flujo básico		
Personal de salud	Sistema	
1. El usuario selecciona la opción generar reporte	El sistema muestra los años disponibles	

RAM por año.	para generar reportes y las RAM.
3. El usuario selecciona el período de tiempo deseado y la RAM.	
4. El usuario pulsa la opción enviar	Muestra el reporte de la cantidad de la reacción seleccionada en el período de tiempo seleccionado.
Flujo Alternativo Si el usuario selecciona el período de tiempo deseado sin la RAM	
3	Muestra el reporte de la cantidad de la reacción en el período de tiempo seleccionado.
Flujo Alternativo Si el usuario selecciona la opción limpiar	
6	Muestra la interfaz nuevamente sin datos.
Flujo Alternativo Si el usuario selecciona la opción home	
7	Ver paso tres del CUS1 Presiona el botón Home para ir a la interfaz menú

Prototipo



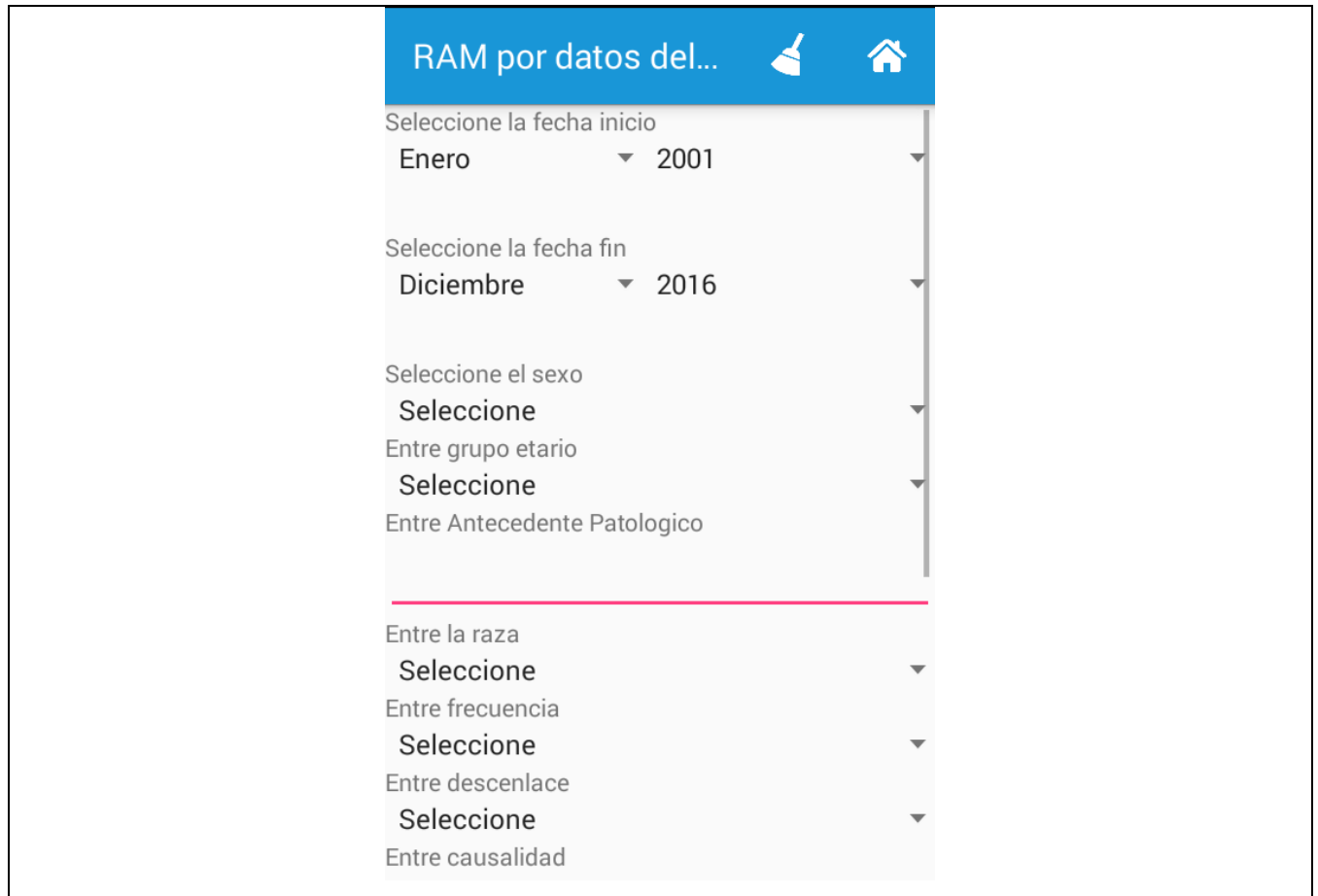
CU3: Generar RAM por medicamento

Objetivo	Generar la cantidad de reacciones adversas a medicamentos por medicamento en un período de tiempo determinado.	
Usuario	Personal de salud	
Resumen	El caso de uso inicia cuando el usuario desea generar un reporte sobre las RAM ocurridas por un medicamento en un período de tiempo determinado.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	Tener información sobre las RAM ocasionadas por ese medicamento en ese período de tiempo.	
Postcondiciones	Reporte generado	
Flujo de eventos		
Flujo básico		
Personal de salud	Sistema	
1. Selecciona la opción generar reporte RAM por medicamento.	El sistema muestra los años disponibles para generar reportes y los medicamentos	
2. Introduce el medicamento y el período de tiempo	Comprueba que el medicamento se encuentre en la base de datos, si el medicamento no se encuentra en la base de datos ir al flujo alterno 4	
3.	Muestra el reporte.	
Flujo Alterno: si el usuario introduce un dato que no se encuentra en la Base de datos		
4	Muestra un mensaje: con los parámetros entrados no se obtuvieron resultados	
Vuelve al paso 3		
Flujo Alterno Si el usuario selecciona la opción limpiar		
6	Muestra la interfaz nuevamente sin datos.	
Flujo Alterno Si el usuario selecciona la opción home		
7	Ver paso tres del CUS1 Presiona el botón Home para ir a la interfaz menú	



CU4: Generar cantidad por datos del paciente

Objetivo	Generar la cantidad de pacientes por los distintos indicadores en un período de tiempo determinado
Usuario	Personal de salud
Resumen	El caso de uso inicia cuando el usuario desea generar un reporte sobre la cantidad de pacientes por indicadores en un período de tiempo determinado.
Complejidad	Media
Prioridad	Alta
Precondiciones	Tener información sobre los datos del paciente.
Postcondiciones	Reporte generado
Prototipo	



CU5: Generar cantidad por datos generales

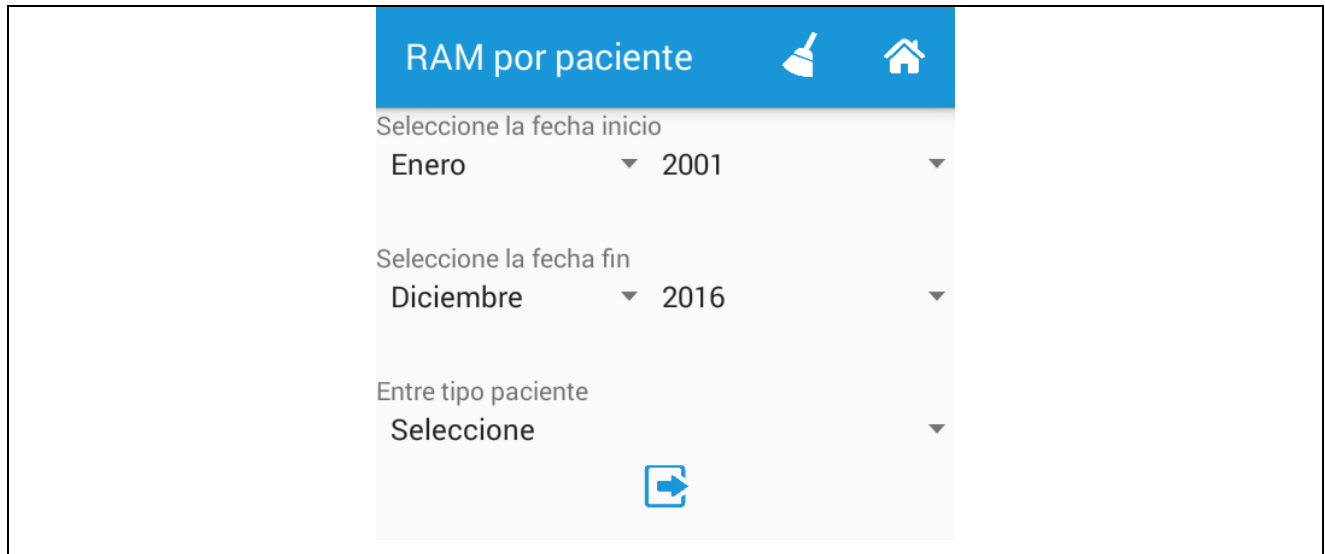
Objetivo	Generar cantidad por datos generales en un período de tiempo determinado
Usuario	Personal de salud
Resumen	El caso de uso inicia cuando el usuario desea generar un reporte sobre la cantidad por datos generales
Complejidad	Media
Prioridad	Alta
Precondiciones	Tener información sobre los datos generales
Postcondiciones	Reporte generado
Prototipo	

CU6: Generar cantidad de RAM por datos registrados

Objetivo	Generar la cantidad de RAM por datos registrados en la base de datos en un período de tiempo determinado
Usuario	Personal de salud
Resumen	El caso de uso inicia cuando el usuario desea generar la cantidad por datos generales
Complejidad	Media
Prioridad	Alta
Precondiciones	Tener información sobre datos generales
Postcondiciones	Reporte generado
Prototipo	

CU7: Generar cantidad de RAM en determinado paciente

Objetivo	Generar cantidad de RAM en determinado paciente
Usuario	Personal de salud
Resumen	El caso de uso inicia cuando el usuario desea generar la cantidad de RAM en determinado paciente
Complejidad	Media
Prioridad	Alta
Precondiciones	Tener información sobre la cantidad de RAM en determinado paciente
Postcondiciones	Reporte generado
Prototipo	



CU8: Porciento de RAM

Objetivo	Graficar el porciento de las RAM en un período determinado
Usuario	Personal de salud
Resumen	El caso de uso inicia cuando el usuario desea generar el porciento de RAM ocurrido en un período de tiempo determinado
Complejidad	Media
Prioridad	Alta
Precondiciones	Tener información sobre el porciento de las RAM
Postcondiciones	Reporte generado
Prototipo	

Porciento de RAM

Seleccione la fecha inicio
Enero 2001

Seleccione la fecha fin
Diciembre 2016

Entre el sexo
Seleccione

Entre la raza
Seleccione

Entre Grupo Etario
Seleccione

Entre la RAM

Entre medicamento

2.5 Conclusiones del capítulo

Luego de realizado el capítulo dos se puede concluir que:

- ✓ El modelo de dominio permitió realizar una propuesta de casos de uso del sistema enfocado a las necesidades de los usuarios finales.
- ✓ La realización del diagrama de casos de uso unido a la descripción textual permitió sentar las bases para desarrollar la propuesta de solución.

Capítulo 3: Implementación y validación de la aplicación móvil Android RamCuba

En el presente capítulo se abordan los temas referentes al diseño, la implementación y prueba de la solución diseñada. Se muestran artefactos generados en la etapa de implementación, además, de los estándares de codificación utilizados en la construcción de la solución. Se detalla el trabajo acerca del tratamiento de errores y se plasma el diagrama de componentes. Se presenta además los resultados de los métodos de validación empleados.

3.1 Patrón Arquitectónico

Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones (Mas, 2013). Android utiliza el patrón arquitectónico Modelo Vista Controlador (MVC), que consiste en separar los datos de una aplicación, la interfaz de usuario y la lógica de negocios en tres componentes distintos que se relacionan para tener como resultado una aplicación.(López et al., 2014)

- ✓ **Modelo:** representación que se construye basada en la información con la que operará una aplicación.
- ✓ **Vista:** interfaz con la que va a interactuar el usuario. En Android, las interfaces se construyen en XML.
- ✓ **Controlador:** actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

A continuación se muestra la Figura 6. Arquitectura modelo vista controlador.

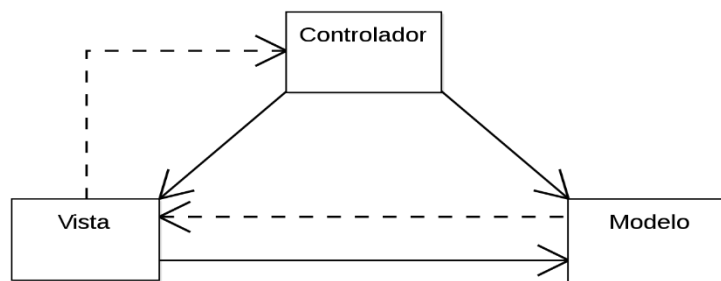


Figura 6: Arquitectura modelo vista controlador
Fuente (Debrauwer, 2013)

3.2 Patrones de diseño

Un patrón de diseño nombra, abstrae e identifica los aspectos clave de un diseño estructurado, común, que lo hace útil para la creación de diseños orientados a objetos reutilizables. Los patrones de diseño identifican las clases participantes y las instancias, sus papeles y colaboraciones, y la distribución de responsabilidades. Cada patrón de diseño se enfoca sobre un particular diseño orientado a objetos. Se describe cuando se aplica, las características de otros diseños y las consecuencias y ventajas de su uso. (Palomo y Gil, 2014b)

3.2.1 Patrones GRASP

Son denominados GRASP (por las siglas en inglés de patrones generales de asignación de responsabilidades), estos se basan en la determinación de las clases adecuadas y decidir cómo estas clases deben interactuar (Cibbarelli, 1999). A continuación, se presentan los patrones GRASP empleados y su aplicación en el diseño.

Patrón Experto

El experto en información es el encargado de definir las interacciones básicas entre los objetos, asignando responsabilidades a las clases de forma tal que el sistema sea más fácil de entender, mantener y ampliar, permitiendo reutilizar los componentes creados en futuras aplicaciones (Feinberg, 2001). Es necesario asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Se utiliza en la aplicación en la realización del paquete arreglos donde cada clase maneja la información referente en la base de datos. Un ejemplo de la utilización de este patrón se evidencia en la clase Medicamentos .java que maneja los nombres y los IDs de los medicamentos de la base de datos.

Patrón Creador

El propósito fundamental de este patrón es asignar responsabilidades relacionadas con la creación de objetos producidos en cualquier evento. Indica que una clase es idónea para asumir la responsabilidad de crear el elemento registrado o contenido (López et al., 2014). Una clase es responsable de crear una instancia de otra clase si: agrega objetos, contiene referencias a objetos, utiliza estrechamente los objetos o tiene la información de inicialización que se necesita para crear un objeto de esa clase. Facilita un Bajo Acoplamiento, supone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización. Este patrón se evidencia en el *MainActivity* que tiene la responsabilidad de crear las demás *activitys* y las clases con la información a mostrar al usuario.

Patrón Alta cohesión

Asigna a las clases responsabilidades que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad, evitando así que una clase sea la única responsable de muchas tareas en áreas funcionales muy heterogéneas. La información que maneja una entidad de software tiene que estar conectada lógicamente con esta, no deben existir entidades con atributos que describan comportamientos que en realidad no le corresponden.(Muñoz, Velthuis y Rubia, 2013). Para verificar que el patrón está presente en la aplicación se aplicó la métrica Relaciones entre clases.

Patrón Bajo acoplamiento

Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad. No puede considerarse en forma independiente como Experto o Alta Cohesión. El acoplamiento tal vez no sea tan importante, sino se busca la reutilización. No existe una medida absoluta de cuando el acoplamiento es excesivo. En términos generales, han de tener escaso acoplamiento las clases muy genéricas y con grandes probabilidades de reutilización (Muñoz, Velthuis y Rubia, 2013). Para verificar que el patrón está presente en la aplicación se aplicó la métrica Relaciones entre clases.

Patrón Controlador

El patrón controlador sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento. (Muñoz, Velthuis y Rubia, 2013)Se utiliza en todas las *activity*⁹ donde se muestra la vista a través del XML y se utilizan los valores a través de la clase controladora en java.

⁹ Una Activity es un componente de la aplicación que contiene una pantalla con la que los usuarios pueden interactuar para realizar una acción.

Patrón Observador:

Define una dependencia de uno-a-muchos entre objetos, de tal forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él (Larman, 2004). Se evidencia en la captura de los eventos asociados a los métodos *listener* de los controles de Android.

3.2.2 Patrones de diseño GOF

Los patrones de diseño GOF, son clasificados según su propósito en creacionales, estructurales y de composición, mientras que respecto a su ámbito se clasifican en clases y objetos. (Muñoz, Velthuis y Rubia, 2013)

Según su propósito:

1. De Creación: resuelven problemas relativos a la creación de objetos.
2. Estructurales: resuelven problemas relativos a la composición de objetos.
3. De Comportamiento: resuelven problemas relativos a la interacción entre objetos.

Según su ámbito:

1. Clases: relaciones estáticas entre clases.
2. Objetos: relaciones dinámicas entre objetos.

Entre los existentes se utilizó en la solución del problema el patrón Singleton, el cual se puede emplear en proyectos simples para contener todo el código bajo una única variable global. En grandes proyectos, puede ser usado para agrupar código relacionado para simplificar la permanencia del mismo y localizarlo en un único lugar, facilitando la modularidad. Incluso en los proyectos de mayor envergadura, Singleton ayuda a optimizar la aplicación, facilitando la carga asíncrona de partes del código que no sean usadas frecuentemente (Muñoz, Velthuis y Rubia, 2013). Este patrón es utilizado en la clase Visualizador.

3.3 Métricas para la verificación del diseño de clases

Esta métrica se aplica con el objetivo de validar el nivel de reutilización y responsabilidad de las clases y la complejidad de implementación, así como el nivel de acoplamiento y la complejidad de mantenimiento. Dentro del contexto de la ingeniería del software, una medida proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto. El

IEEE¹⁰ define métrica como una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado (Durango, 2015a). Las métricas suelen calificarse como instrumentos que cuantifican un criterio determinado. Estas son utilizadas para indicar la calidad de un producto de software, evaluar los beneficios en términos de productividad y de calidad, así como para justificar el uso de determinadas herramientas, entre otras. (Portilla, 2013)

Las métricas concebidas como instrumento para evaluar la calidad del diseño y la relación con los atributos de calidad de la investigación fueron:

- ✓ Tamaño Operacional de clases (TOC).
- ✓ Relaciones entre clases (RC).

3.3.1 Tamaño Operacional de Clase (TOC)

El tamaño general de una clase puede medirse determinando las siguientes medidas: (Portilla, 2013)

- ✓ El total de operaciones (tanto heredadas como privadas de la instancia), que se encapsulan dentro de la clase.
- ✓ El número de atributos (tanto heredados como privados de la instancia), encapsulados por la clase.
- ✓ Valores grandes de TOC representan gran responsabilidad de la clase. Esto implica la reducción de la reutilización de la clase y complica la implementación y las pruebas. De forma general, operaciones y atributos deben ser ponderados al determinar el tamaño de la clase. Para valores pequeños de TOC para una clase existe mayor posibilidad de que la clase pueda ser reutilizada.

Valores grandes de TOC representan gran responsabilidad de la clase. Esto implica la reducción de la reutilización de la clase y complica la implementación y las pruebas. De forma general, operaciones y atributos deben ser ponderados al determinar el tamaño de la clase. Para valores pequeños de TOC para una clase existe mayor posibilidad de que la clase pueda ser reutilizada.

¹⁰ IEEE: Instituto de Ingeniería Eléctrica y Electrónica: asociación mundial de ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas.

Tabla 5 Métrica Tamaño operacional de las clases
Fuente (Martínez, 2017b)

Parámetros de Calidad	Valores Grandes de TOC
Reutilización	Reduce la reutilización de la clase
Implementación	Complica la implementación
Responsabilidad	La clase debe tener bastante responsabilidad

Medidas y umbrales aplicados:

Tabla 6 Umbrales utilizados en TOC
Fuente (Martínez, 2017b)

Parámetros	Criterio	Umbral
Responsabilidad	Pequeño	\leq Prom.
	Medio	Entre Prom. y 2^* Prom.
	Grande	$> 2^*$ Prom.
Complejidad de Implementación	Baja	\leq Prom.
	Media	Entre Prom. y 2^* Prom.
	Alta	$> 2^*$ Prom.
Reutilización	Baja	$> 2^*$ Prom.
	Media	Entre Prom. y 2^* Prom.
	Alta	\leq Prom.

Se les aplicó la métrica de TOC a un total de 23 clases para un total de 54 atributos, obteniéndose un promedio de atributos de 2.34 y a un total de 31 operaciones con un promedio de operaciones de 1.35. De las clases analizadas se tienen que, de acuerdo con la responsabilidad, 18 tienen responsabilidad baja, 2 media y 3 alta; respecto a la complejidad 15 clases la tienen baja, 8 media y 0 alta; y referente a la reutilización 3 la presentan baja, 2 media y 18 alta. Este resultado puede verse reflejado en el gráfico de barra presentado a continuación, (ver Figura 7 Resultado de la métrica TOC).

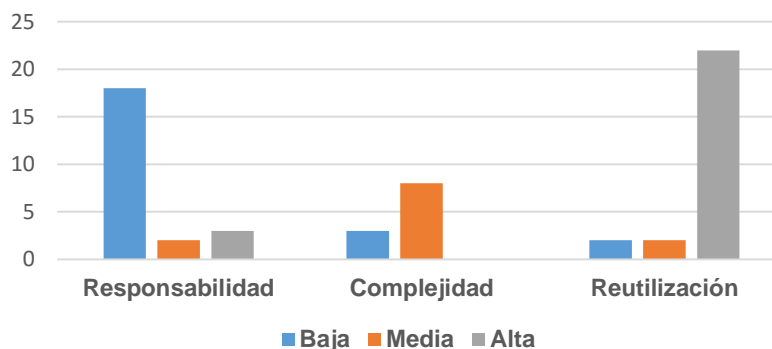


Figura 7 Resultado de la métrica TOC
Fuente: elaboración propia

CAPÍTULO 3: Implementación y validación de la aplicación móvil Android RamCuba

A partir de estos resultados, se concluye que la complejidad y la responsabilidad del diseño realizado son generalmente bajas; no se presentarán problemas con la implementación y las clases dentro del sistema se podrán reutilizar.

3.3.2 Relaciones entre clases (RC)

Está dado por el número de relaciones de uso de una clase con otras. Los atributos de calidad que esta afectan son Acoplamiento, Complejidad de mantenimiento, Cantidad de Pruebas y Reutilización.

Tabla 7 Métrica Relaciones entre clases
Fuente (Durango, 2015b)

Parámetros de Calidad	Valores grandes de RC
Acoplamiento	Implica un aumento del acoplamiento de la clase
Complejidad de mantenimiento	Implica un aumento de la complejidad del mantenimiento de la clase.
Reutilización	Implica una disminución en el grado de reutilización de la clase.
Cantidad de Pruebas	Implica un aumento de la cantidad de pruebas de unidad necesarias para probar una clase.

Medidas y umbrales aplicados:

Tabla 8 Umbrales utilizados en RC
Fuente (Durango, 2015b)

Parámetros	Criterio	Umbral
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
Complejidad de mantenimiento	Baja	\leq Prom
	Media	Entre Prom. y $2 \cdot$ Prom
	Alta	$> 2 \cdot$ Prom.
Reutilización	Baja	$> 2 \cdot$ Prom.
	Media	Entre Prom. y $2 \cdot$ Prom
	Alta	\leq Prom
Cantidad de Pruebas	Baja	\leq Prom
	Media	Entre Prom. y $2 \cdot$ Prom.
	Alta	$> 2 \cdot$ Prom.

Una vez aplicada la métrica RC, se obtiene un total de 22 relaciones de uso para 23 clases, siendo un promedio de 0.95 relaciones. Con estas relaciones se determina que el acoplamiento es generalmente bajo en las clases, al igual que la complejidad de mantenimiento y la cantidad de pruebas, además, de

que la reutilización de las clases es alta. Este resultado puede verse reflejado en el gráfico de barra presentado a continuación, (ver Figura.8 Resultado de la métrica RC).

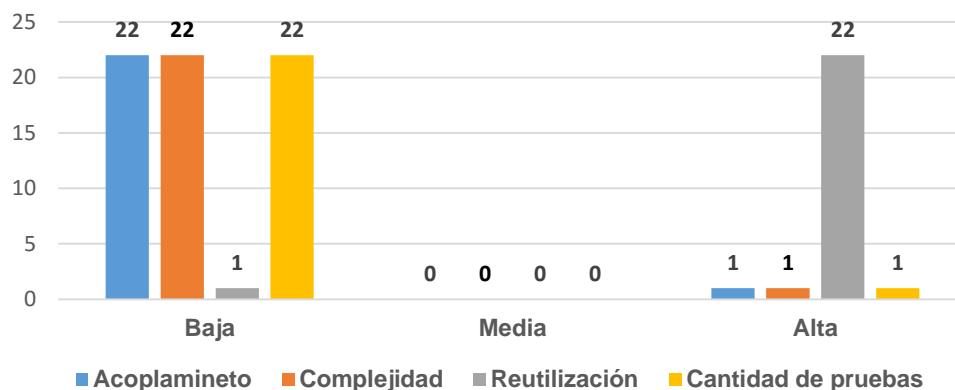


Figura 8 Resultado de la métrica RC

Fuente: elaboración propia

Para aplicar estas métricas se utilizó el diagrama de clases ([ver anexo 3](#)).

3.4 Modelo de datos

El modelado de datos es el lenguaje empleado para realizar la representación de una base de datos. Se define modelo de datos como el conjunto de herramientas conceptuales para describir la representación de la información en términos de datos. Los modelos de datos comprenden aspectos relacionados con: estructuras y tipos de datos, operaciones y restricciones (GÓMEZ, 2012). Para la solución se utiliza una base de datos relacional en SQLite que es el sistema de gestión de bases de datos relacional que soporta el sistema operativo Android. La base de datos en SQLite es una migración del mercado de datos del producto Synta. Luego del análisis realizado en el epígrafe [1.3](#) al Mercado de datos del producto Synta se hace necesario realizar una limpieza de los datos eliminando las tablas referente a la calidad de la recogida de los datos, así como los id del nomenclador que no serán utilizadas en esta investigación

Para la migración de los datos se utilizó Pentaho Data Integration que permitió aplicar técnicas de cifrado, aumentando así la seguridad e integridad de los datos. Un ejemplo de las transformaciones realizadas en el Pentaho Data Integration es la referente a la tabla dimensión especialidad que se muestra a continuación.

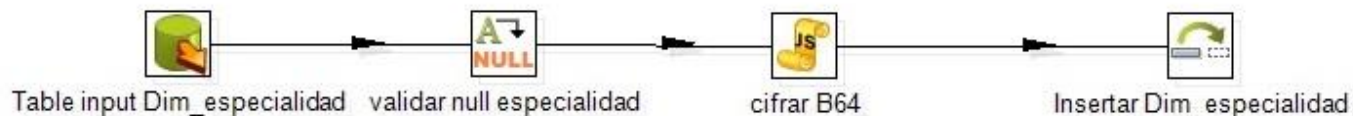


Figura 9 Transformación dimensión especialidad
Fuente: elaboración propia

Como se muestra en la Figura 9 Transformación dimensión especialidad para las transformaciones de los datos se utilizaron varios elementos como son: *Tableinput* que permite establecer la conexión con el origen de los datos y seleccionar los datos que se deseen extraer, *valuenull* que permite redefinir los elementos null del origen (se sustituyen por “NO REFIERE”), *Javascript modificado* que permite codificar los datos entrados a Base64 y por último un *Insert/update* que permite establecer la conexión con la base de datos destino e introduce los valores. El mismo proceso se utilizó para el resto de tablas. A continuación, se presenta el modelo de datos resultante.

CAPÍTULO 3: Implementación y validación de la aplicación móvil Android RamCuba

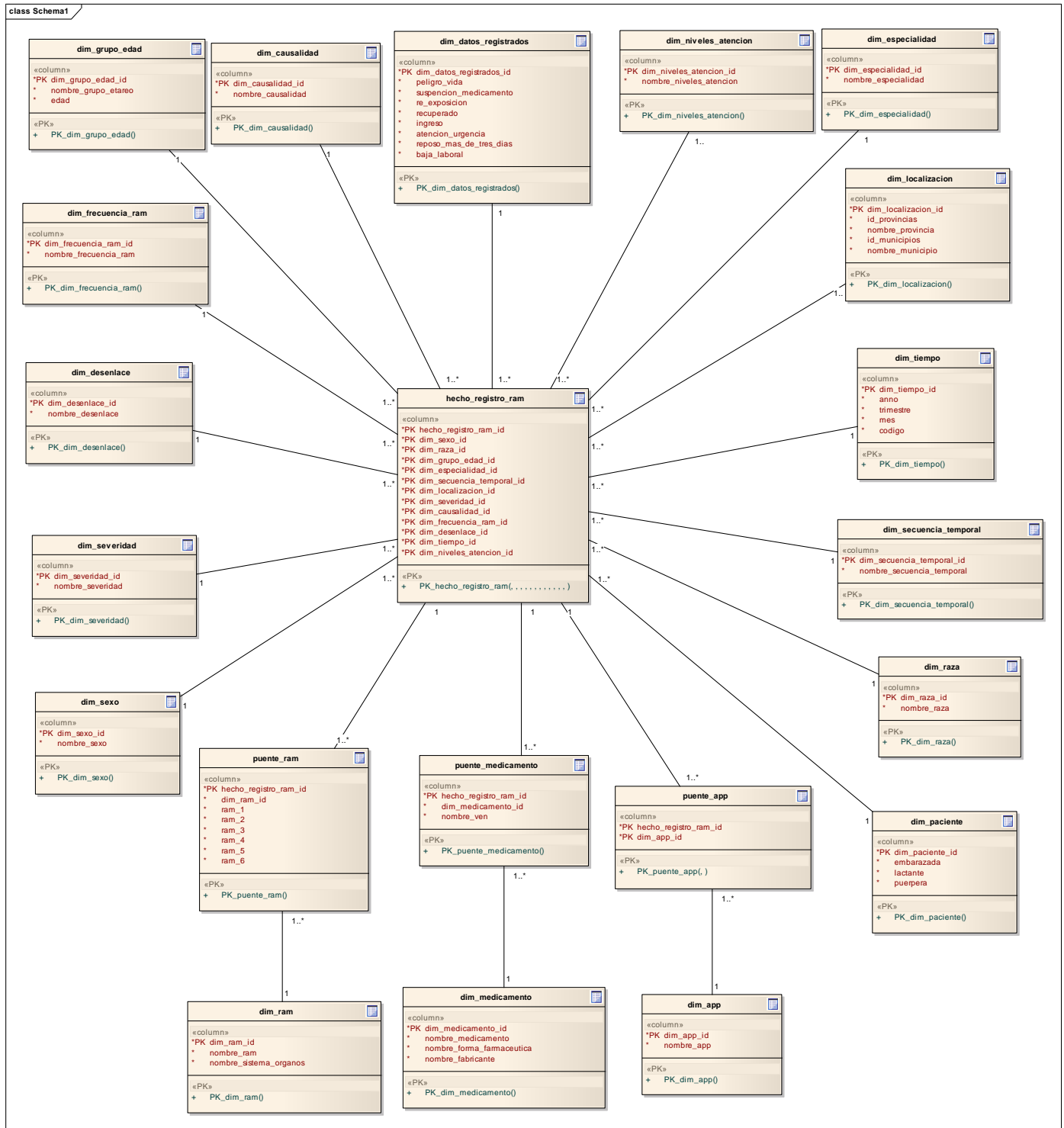


Figura 10 Modelo de datos de la propuesta de solución
Fuente: elaboración propia

Para una mayor comprensión del modelo de datos se describen las tablas de dicho modelo a continuación.

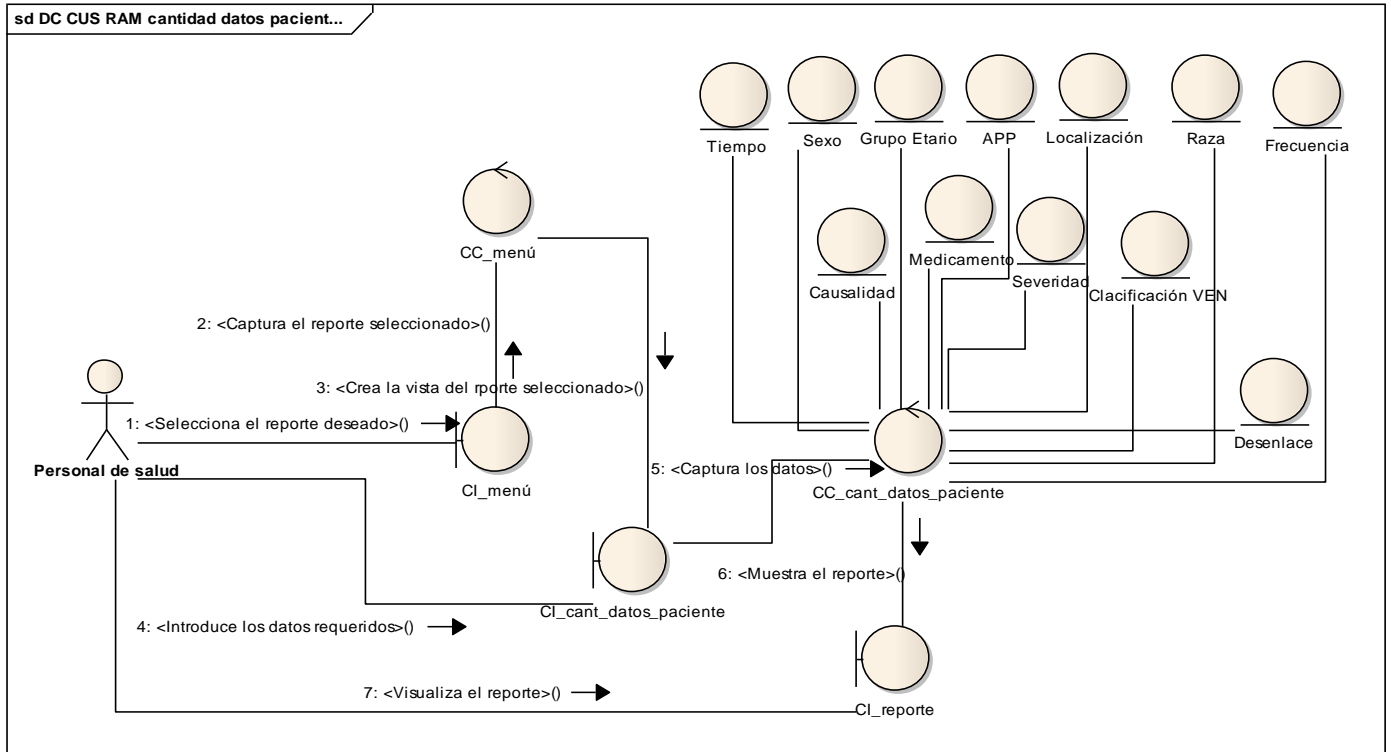


Figura 12 Diagrama de comunicación del CU generar cantidad por datos del paciente
Fuente: elaboración propia

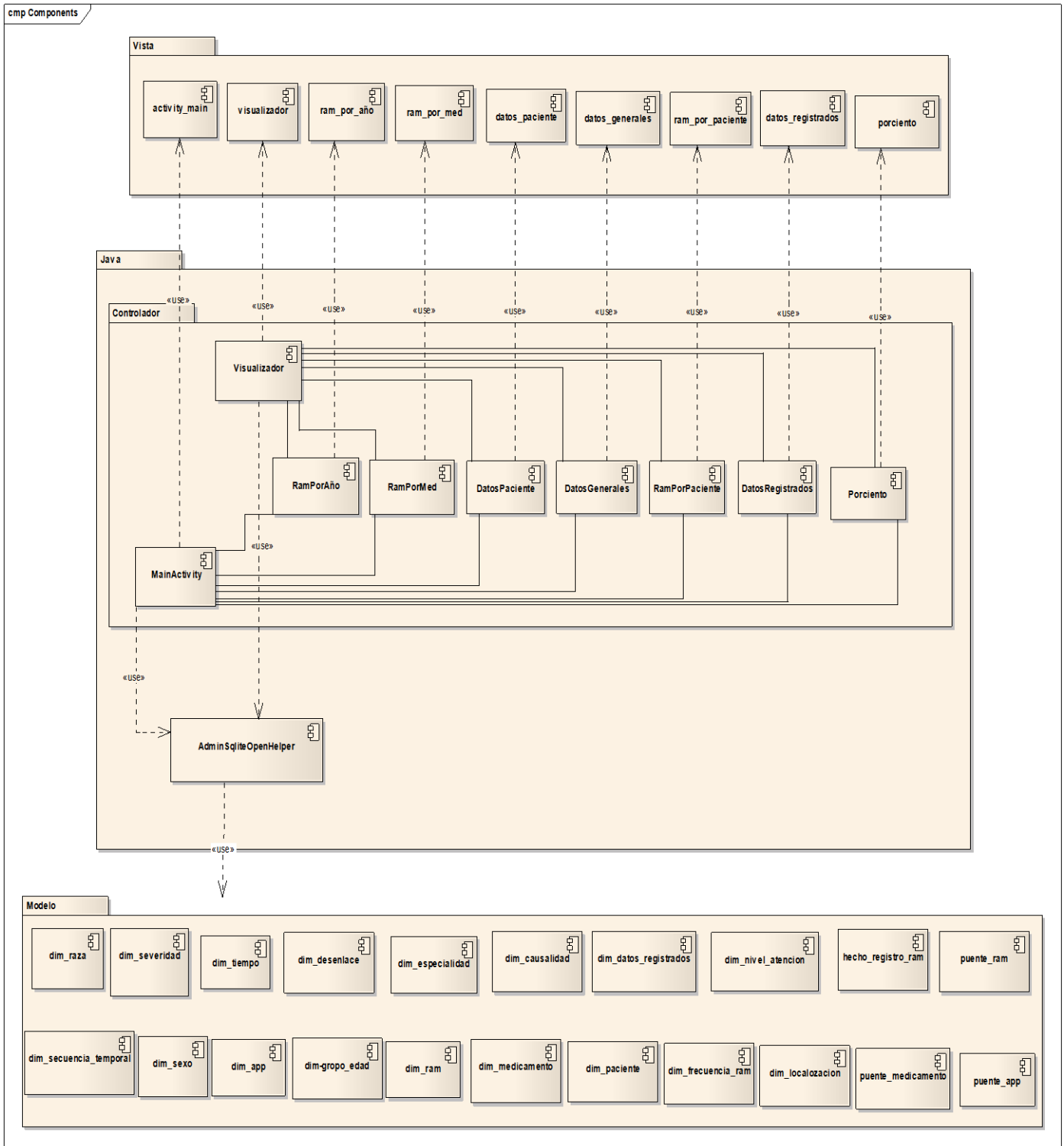


Figura13: Diagrama de componentes
Fuente: elaboración propia

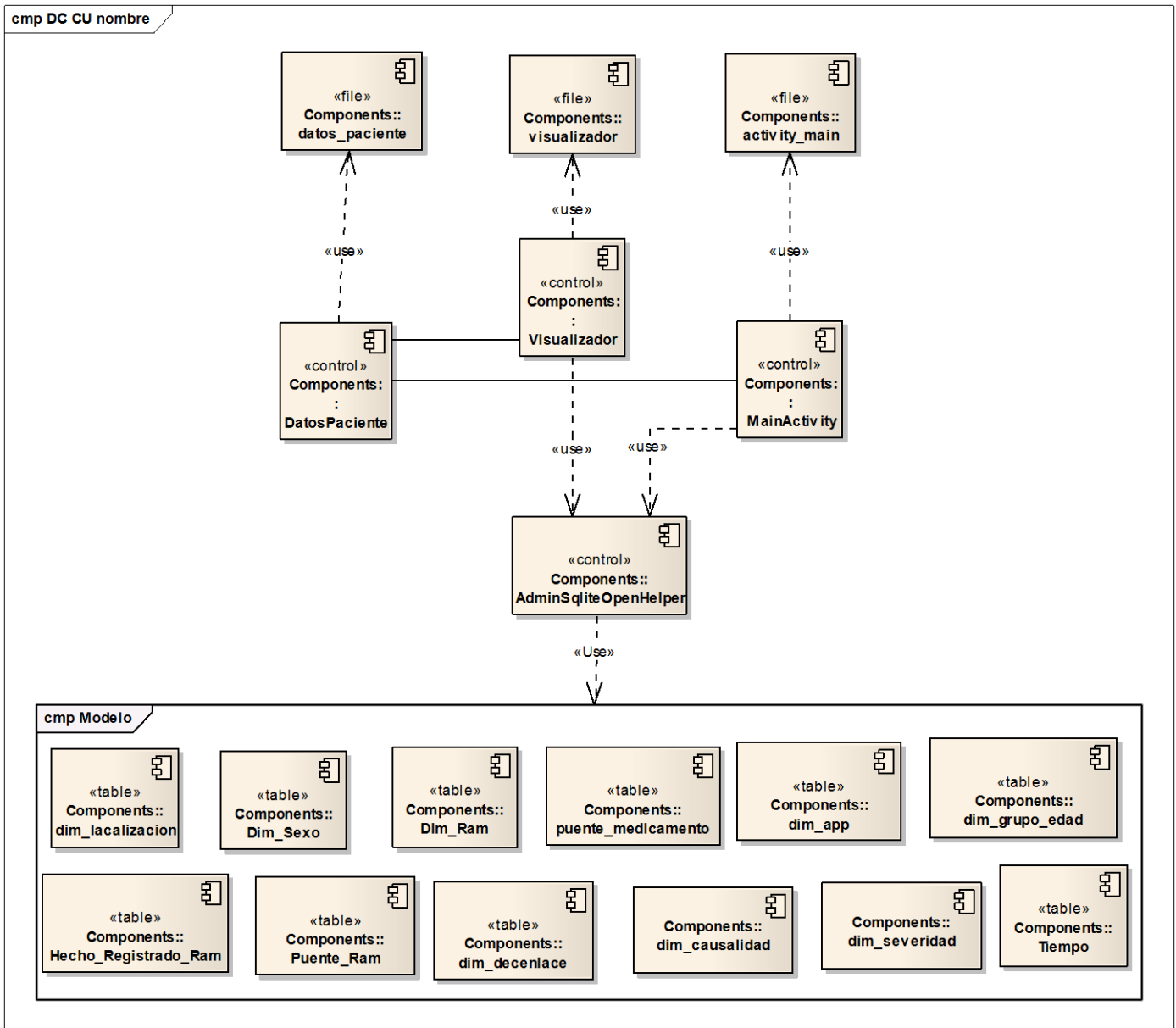


Figura 14: Diagrama de componentes del CU generar cantidad por datos del paciente
Fuente: elaboración propia

3.5.1 Estándar de codificación

Un estándar de codificación comprende todos los aspectos de la generación de código. (Martin, 2012) Un código fuente debe reflejar un estilo ordenado de la implementación de un sistema. Como estándar de codificación para la implementación de la solución se utilizaron:

Nombre de las clases:

- ✓ Los nombres de las clases deben estar expresados en notación UpperCamelCase.
- ✓ No se deben utilizar guiones bajos en su nombre “_”.
- ✓ Deben expresar con claridad cuál es el alcance y la responsabilidad de la clase.
- ✓ Los nombres de las clases no deben estar atados a las clases de las que se derivan, cada clase debe tener un significado por ella misma, no en dependencia de la clase de la que deriva.

Ejemplo:

RamAnno, RamMedicamento, etc.

Nombre de las funciones:

- ✓ Todas las funciones definidas por los desarrolladores deben seguir la nomenclatura CamelCase, a no ser que para cierto ámbito se especifiquen características específicas.
- ✓ Los nombres de las funciones deben dejar reflejado claramente cuál es la acción que realiza.

Ejemplo:

reporteRamAnno, reporteRamMedicamento, etc.

- ✓ Se debe apoyar en los prefijos para expresar la acción que realiza sobre un elemento determinado.

Ejemplo:

Set: para cambiar el valor de una variable.

Get: para devolver un valor determinado.

Política de llaves:

- ✓ Política tradicional de Unix de colocar la primera llave en la misma línea del comienzo del bloque separada por un espacio y la llave de cierre del bloque en la última línea.
- ✓ La estrategia debe ser mantenida en todo el código escrito por el programador y siempre colocar las llaves aunque sea para una sola línea de código dentro del bloque.

Ejemplo:

```
if (condition) { while (condition) {  
... ..  
}}}
```

Políticas de () paréntesis:

- ✓ No utilizar paréntesis seguido de palabras claves del lenguaje, se debe dejar un espacio por el medio.
- ✓ Los paréntesis se sitúan seguidos de los nombres de las funciones.
- ✓ No se debe utilizar paréntesis en las sentencias *return* de no ser necesario.

```
Ejemplo:  
if (condition) { while (condition) {  
  } }  
strcmp(s, s1);  
return 1;
```

Políticas de Sangría, Tabulación y Espacios:

- ✓ La sangría de cada nivel será de cuatro espacios o un tabulador.

3.5.2 Tratamiento de errores

En el desarrollo de software el tratamiento de errores es muy importante para garantizar el correcto funcionamiento del sistema, que no afecte la calidad del mismo. Es fundamental identificar y controlar los posibles problemas que puedan presentarse a la hora de interactuar con la aplicación.(Navarro et al., 2013)

Por lo que se diseñó el software de forma tal que el usuario introduzca la menor cantidad posible de datos, evitando así incoherencias e incorrecciones en los mismos. En el caso de la entrada de datos por parte del usuario se implementaron funciones que validaron dicha entrada para que, de existir errores, se muestren mensajes de alerta indicando los errores detectados.

3.6 Pruebas de software

Las pruebas de software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación (Sommerville, 2011b). Una vez terminada la implementación del sistema, es necesario probar el software con el objetivo de identificar y corregir el máximo de errores posibles antes de ser entregados al cliente. En la presente investigación se aplicó las pruebas de caja negra, con la técnica de partición equivalente que se explican a continuación.

3.6.1 Pruebas de caja negra

La prueba de caja negra pretende demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta. Se centran en los requisitos funcionales del software y permiten encontrar errores de tipo de funciones incorrectas o ausentes, de interfaz, en la estructura de datos, de rendimiento y de inicialización y terminación. (Pressman, 2010)

Para la realización de este método se utilizó la técnica de partición equivalente, la cual divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba ([ver anexo 4](#)) que hay que desarrollar. (Pressman, 2010)

En la fase de pruebas se realizaron tres iteraciones a la aplicación, en la primera se detectaron cuatro no conformidades. Se realizó posteriormente una segunda iteración de pruebas para verificar que se habían eliminado los problemas encontrados en estos requisitos o si se encontraban nuevas no conformidades en el funcionamiento de la aplicación donde se encontraron tres no conformidades. Esta tercera iteración concluyó satisfactoriamente ya que no se encontró ningún otro problema en la aplicación.

3.6.2 Prueba de Rendimiento

Las **pruebas de rendimiento** son las pruebas que se realizan, desde una perspectiva, para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo (Medina, 2014). Se le realizó pruebas de rendimiento agregándole 130355 *tuplas* de información de prueba a la base de datos, obteniéndose los siguientes resultados:

Tabla 9 Resultado de la prueba de rendimiento
Fuente elaboración propia

Memoria RAM del dispositivo móvil	Procesador del dispositivo móvil	Al cargar los datos	Al realizar de reportes
512 MB	1GHz Dual-Core	6 segundos	4 segundos
1 GB	1.2 GHz Dual-Core	5 segundos	3 segundos

Los resultados obtenidos en esta prueba pueden calificarse como satisfactorios ya que en el peor de los casos ronda los seis segundos, tiempo aceptado por la mayoría de los usuarios que probaron la aplicación.

3.7 Aplicación de la técnica IADOV para medir satisfacción

Se aplicó la técnica de **IADOV** (Sosa et al., 2012) (Fabre y Padrón, 2014) (Sotelo y Arvelo, 2015) para obtener el grado de satisfacción de los usuarios respecto a la aplicación RamCuba. Los criterios utilizados

CAPÍTULO 3: Implementación y validación de la aplicación móvil Android RamCuba

se fundamentan en las relaciones que se establecen entre tres preguntas cerradas, que se intercalan dentro de la encuesta aplicada. Para obtener los resultados de la aplicación de la técnica, es necesario conocer la escala de satisfacción, así como la fórmula para determinar la satisfacción del grupo. La escala de satisfacción personal e individual para estas actividades responde a la siguiente estructura, en función de la puntuación obtenida:

La escala de satisfacción

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definida
4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictoria

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma:

- | | |
|------|-------------------------------------|
| +1 | Máximo de satisfacción (A) |
| 0,5 | Más satisfecho que insatisfecho (B) |
| 0 | No definido y contradictorio (C) |
| -0,5 | Más insatisfecho que satisfecho (D) |
| -1 | Máxima insatisfacción (E) |

La satisfacción grupal se calcula por la siguiente fórmula:

$$ISG = A(+1) + B(+0,5) + C(0) + D(-0,5) + E(-1)/N$$

CAPÍTULO 3: Implementación y validación de la aplicación móvil Android RamCuba

En la Tabla 10 se muestra el *Cuadro lógico de IADOV*, donde se realiza la triangulación de las preguntas directas.

Tabla 10 Cuadro Lógico de IADOV
Fuente: elaboración propia

	1. ¿Considera Ud. importante utilizar esta aplicación?								
	No			No sé			Sí		
	3. Si tuvieras la aplicación RamCuba ¿La usaría?								
4. ¿Le satisface los resultados de los reportes de RAM, (tanto gráficos como nominales) que muestra RamCuba?	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
No me gusta tanto	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

El índice de satisfacción grupal se encuentra entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción.

Para medir la satisfacción se le aplicó la encuesta ([ver anexo 5](#)) a una muestra de 57 profesionales del Hospital General Docente Danil Codorniu Pruna de Placetas y en el Hospital Provincial Camilo Cienfuegos en Sancti Spíritus distribuidos de la siguiente forma:

Cantidad de profesionales por años de experiencia

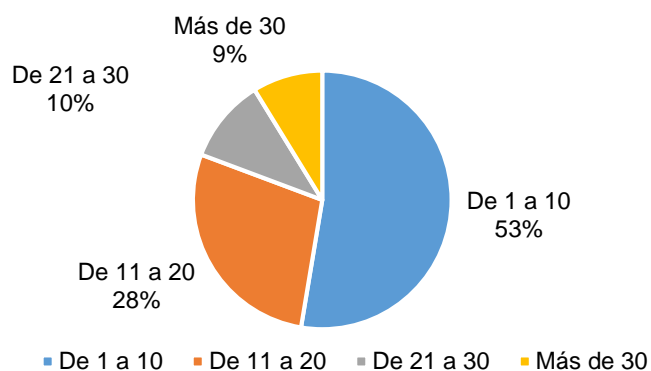


Figura 13 Distribución de profesionales por años de experiencia
Fuente: elaboración propia



*Figura 14 Figura Distribución de profesionales por especialidad
Fuente: elaboración propia*

El ISG al aplicar la técnica IADOV fue 0.66, este valor al encontrarse entre el intervalo de 0,5 a +1 indica que existe máxima satisfacción por parte de los usuarios potenciales de la aplicación RamCuba.

3.8 Conclusiones del capítulo

Luego de realizado el capítulo tres se puede concluir que:

- ✓ Al aplicar las métricas Tamaño Operacional de Clases y Relaciones entre Clases, se obtuvo que existe un bajo acoplamiento y alta cohesión entre las clases de la aplicación, demostrando que futuros cambios serían de forma sencilla.
- ✓ Los estándares de codificación aplicados en el desarrollo de la aplicación permitió obtener un código fuente estándar y fácil de entender.
- ✓ Las pruebas de caja negra aplicadas demostraron que se implementaron correctamente los requisitos funcionales definidos en el desarrollo de la aplicación.

Conclusiones

- ✓ A través del análisis de soluciones existentes, se determinó que ninguna podría ser utilizada para la solución del problema, por lo que se hizo necesario la implementación de una nueva aplicación teniendo cuenta elementos similares con las herramientas analizadas.
- ✓ El análisis del mercado de datos de Synta permitió identificar la estructura necesaria de la base de datos, así como delimitar la información necesaria de la aplicación a desarrollar.
- ✓ Al aplicar el método de Bohem y Turner quedó demostrado que el enfoque de la metodología a utilizar es ágil.
- ✓ La selección de las herramientas, tecnologías y metodologías de desarrollo contribuyó a agilizar la implementación de la aplicación RamCuba, teniendo en cuenta la poca experiencia del equipo de desarrollo.
- ✓ La estrategia de validación aplicada permitió probar que la aplicación RamCuba cumple tanto con estándares informáticos como con la aprobación de los futuros usuarios potenciales.

Recomendaciones

Una vez concluido el desarrollo de la aplicación RamCuba y cumplido el objetivo general de la investigación, se recomienda:

- ✓ Adicionar a la aplicación RamCuba una funcionalidad que le permita al médico reportar incidencias de RAM en cualquier momento desde su dispositivo móvil.

Referencias bibliográficas

1. AARP, fuente:, 2015. Efectos secundarios de medicamentos comunes - AARP en español. *AARP* [en línea]. [Consulta: 28 febrero 2017]. Disponible en: <http://www.aarp.org/espanol/salud/farmacos-y-suplementos/info-2015/efectos-secundarios-medicamentos-comunes-fotos.html>.
2. AMO, F.A., NORMAND, L.M. y PÉREZ, F.J.S., 2013. *Introducción a la ingeniería del software*. S.I.: Delta Publicaciones. ISBN 978-84-96477-00-1.
3. AYALA, L., 2015. *Android Studio Curso Básico: Aprenda Paso a Paso*. S.I.: CreateSpace Independent Publishing Platform. ISBN 978-1-5173-7273-6.
4. BELLO, A.S. y LÓRIGA, D.S., 201d. C. *Desarrollo de un Data Mart para la toma de decisiones referente a las Reacciones Adversas a Medicamentos, en el Ministerio de Salud Pública, desde el producto Synta*. Habana: Universidad de las Ciencias Informáticas.
5. BURGUÉS, E.G., 2016. *Aprende a Modelar Aplicaciones con UML: 2ª Edición*. S.I.: IT Campus Academy. ISBN 978-1-5234-9853-6.
6. CIBBARELLI, P., 1999. *Proceedings of the Fourteenth IOLS '99*. S.I.: Information Today, Incorporated. ISBN 978-1-57387-085-6.
7. DEBRAUWER, L., 2013. *Patrones de diseño en Java: Los 23 modelos de diseño: descripción y solución ilustradas en UML 2 y Java*. S.I.: Ediciones ENI. ISBN 978-2-7460-8645-6.
8. DURANGO, A., 2015a. *Diseño de Software: 2ª Edición*. S.I.: IT Campus Academy. ISBN 978-1-5196-2073-6.
9. DURANGO, A., 2015b. *Diseño de Software: 2ª Edición*. S.I.: IT Campus Academy. ISBN 978-1-5196-2073-6.
10. ERRECALDE, J.O., 2012. La evolución del medicamento en la historia. [en línea], [Consulta: 27 febrero 2017]. ISSN 0327-8093. Disponible en: <http://hdl.handle.net/10915/27667>.
11. FABRE, A.F. de C. y PADRÓN, A.L., 2014. Validación mediante criterio de usuarios del sistema de indicadores para prever, diseñar y medir el impacto en los proyectos de investigación del sector agropecuario. *Revista Ciencias Técnicas Agropecuarias* [en línea], vol. 23, no. 3. [Consulta: 28 septiembre 2016]. ISSN 2071-0054. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2071-00542014000300012.
12. FEINBERG, R., 2001. *The Changing Culture of Libraries: How We Know Ourselves Through Our Libraries*. S.I.: McFarland. ISBN 978-0-7864-5020-6.
13. FNM, 2017. Formulario Nacional de Medicamentos. CUBA. - Home. [en línea]. [Consulta: 20 mayo 2017]. Disponible en: <http://fnmedicamentos.sld.cu//index.php?P=Home>.

14. GIRONÉS, J.T., 2012. *El Gran Libro de Android*. S.I.: Marcombo. 1ª. S.I.: MARCOMBO, S.A. ISBN 978-84-267-1732-0.
15. GÓMEZ, J.M.P., 2012. *Bases de datos relacionales y modelado de datos*. S.I.: Ediciones Paraninfo, S.A. ISBN 978-84-283-3356-6.
16. KUHRMANN, M., MÜNCH, J. y RAUSCH, A., 2016. *Managing Software Process Evolution*. S.I.: s.n. ISBN 978-3-319-31545-4.
17. LEÓN, R.A.H. y GONZÁLEZ, S.C., 2011. *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA Rolando Alfredo Hernández León y Sayda Coello González*. Hernández León, Rolando Alfredo. Ciudad de La Habana: Editorial Universitaria del Ministerio de Educación Superior, 2011. ISBN 978-959-16-1307-3.
18. LÓPEZ, E.T., RAMON, A.O., SARROCA, E.M. y SEONE, C.G., 2014. *Diseño de sistemas software en UML*. S.I.: Univ. Politèc. de Catalunya. ISBN 978-84-9880-075-3.
19. LÓPEZ, G.J. y ORTA, A.I., 2015. *Normas y procedimientos de trabajo del Sistema Cubano de Farmacovigilancia*. La Habana: s.n.
20. MARTÍN, J.D., 2015. Aplicación Android del sistema de evaluación de contenidos: e-Liza. [en línea], [Consulta: 28 febrero 2017]. Disponible en: <http://uvadoc.uva.es:80/handle/10324/15113>.
21. MARTIN, R.C., 2012. *Código limpio: manual de estilo para el desarrollo ágil de software*. S.I.: Anaya Multimedia-Anaya Interactiva. ISBN 978-84-415-3210-6.
22. MARTÍNEZ, R.N., 2017a. *El Proceso de Desarrollo de Software: 2ª Edición*. S.I.: IT Campus Academy. ISBN 978-1-5428-6046-8.
23. MARTÍNEZ, R.N., 2017b. *El Proceso de Desarrollo de Software: 2ª Edición*. S.I.: IT Campus Academy. ISBN 978-1-5428-6046-8.
24. MAS, A., 2013. *Agentes software y sistemas multiagente: conceptos, arquitecturas y aplicaciones*. S.I.: Pearson Educación. ISBN 978-84-205-4367-3.
25. MEDINA, J., 2014. *Pruebas de Rendimiento TIC*. S.I.: Lulu.com. ISBN 978-1-291-93438-0.
26. MORALES, A.A., DANNIER, F.R., ROLANDO, P.G., JOSÉ, M.A. y GUILLERMO, S.L., 2012. SLD137 SISTEMA INTEGRAL PARA EL CONTROL FARMACOLÓGICO. *Informática Salud 2013* [en línea]. S.I.: s.n., [Consulta: 28 febrero 2017]. Disponible en: <http://www.informatica2013.sld.cu/index.php/informaticasalud/2013/paper/view/296>.
27. MORALES, A.A., PÉREZ, D.C. y LORENZO, G.S., 2012. SLD136 SISTEMA PARA LA RECOLECCIÓN Y CONTROL DE LAS REACCIONES ADVERSAS A MEDICAMENTOS OCURRIDAS EN CUBA. *Informática Salud 2013* [en línea]. S.I.: s.n., [Consulta: 20 mayo 2017]. Disponible en: <http://www.informatica2013.sld.cu/index.php/informaticasalud/2013/paper/view/295>.
28. MUÑOZ, C.C., VELTHUIS, M.G.P. y RUBIA, M.Á.M. de la, 2013. *Calidad del producto y proceso software*. S.I.: Editorial Ra-Ma.

29. NAVARRO, F.M., CORRALES, J.D., VILLAR, M.L.G., TORRE, A.S.D.L., CIFREDO, E.A.P. y OLIVAS, L.M., 2013. *Técnicos de soporte informático de la comunidad de castilla y león. Temario volumen i*. S.I.: MAD-Eduforma. ISBN 978-84-665-5098-7.
30. OMS, 2011a. Formulario Modelo de la OMS 2011: Consejo general a los prescriptores: Efectos adversos e interacciones. [en línea]. [Consulta: 27 febrero 2017]. Disponible en: <http://apps.who.int/medicinedocs/es/d/Js5422s/4.4.html>.
31. OMS, 2011b. OMS | La OMS insiste en la necesidad de velar por la seguridad de los medicamentos de uso pediátrico. *WHO* [en línea]. [Consulta: 27 febrero 2017]. Disponible en: <http://www.who.int/mediacentre/news/releases/2007/pr51/es/>.
32. OMS, 2017. OMS | Personal sanitario. *WHO* [en línea]. [Consulta: 1 mayo 2017]. Disponible en: http://www.who.int/topics/health_workforce/es/.
33. ORTA, I.A. y LÓPEZ, G.J., 2015. *Normas y procedimientos de trabajo del sistema cubano de farmacovigilancia*. S.I.: MINSAP.
34. ORTEGA, D.L., VILLA, S. y ANDREA, J., 2012. *Estudio comparativo de las herramientas CASE: Staruml, Poseidon For UML y Enterprise Architect, para el modelamiento de diagramas UML* [en línea]. Thesis. S.I.: Universidad Tecnológica de Pereira. [Consulta: 28 febrero 2017]. Disponible en: <http://repositorio.utp.edu.co/dspace/handle/11059/2737>.
35. PALOMO, S.R.G. y GIL, E.M., 2014a. *Aproximación a la ingeniería del software*. S.I.: Editorial Universitaria Ramon Areces. ISBN 978-84-9961-093-1.
36. PALOMO, S.R.G. y GIL, E.M., 2014b. *Aproximación a la ingeniería del software*. S.I.: Editorial Universitaria Ramon Areces. ISBN 978-84-9961-093-1.
37. PÉREZ, D.C. y LORENZO, G.S., 2012. SLD136 SISTEMA PARA LA RECOLECCIÓN Y CONTROL DE LAS REACCIONES ADVERSAS A MEDICAMENTOS OCURRIDAS EN CUBA. *Informática Salud 2013* [en línea]. S.I.: s.n., [Consulta: 1 mayo 2017]. Disponible en: <http://www.informatica2013.sld.cu/index.php/informaticasalud/2013/paper/view/295>.
38. PÉROCHON, S.H.-S., 2014. *Android: Guía de desarrollo de aplicaciones para Smartphones y Tablet* (2a edición). S.I.: ENI. ISBN 978-2-7460-9229-7.
39. PORTILLA, J.R.C., 2013. *Calidad educativa en ingeniería de sistemas*. S.I.: Universidad del Norte. ISBN 978-958-741-351-9.
40. PRESSMAN, R.S., 2010. *Ingeniería del Software: Un Enfoque Práctico*. 7ma Edición. S.I.: s.n. ISBN 978-607-15-0314-5.
41. RAE- ASALE, 2014a. Diccionario de la lengua española - Edición del Tricentenario. *Diccionario de la lengua española* [en línea]. [Consulta: 28 febrero 2017]. Disponible en: <http://dle.rae.es/?id=OkIjC3R>.

REFERENCIAS BIBLIOGRÁFICAS

42. RAE- ASALE, 2014b. Diccionario de la lengua española - Edición del Tricentenario. *Diccionario de la lengua española* [en línea]. [Consulta: 28 febrero 2017]. Disponible en: <http://dle.rae.es/?id=OkIjC3R>.
43. SALA, J.J.R., 2013. *Introducción a la programación. Teoría y práctica: teoría y práctica*. S.I.: Editorial Club Universitario. ISBN 978-84-8454-274-2.
44. SALUD, R.U.D.T.M.D., 2011. THE UNITED REPUBLIC OF TANZANIA MINISTRY OF HEALTH GUIDELINES FOR MONITORING AND REPORTING ADVERSE DRUG REACTIONS (ADRs). [en línea]. [Consulta: 1 mayo 2017]. Disponible en: <http://apps.who.int/medicinedocs/documents/s18571en/s18571en.pdf>.
45. SÁNCHEZ, T.R., 2015. *Metodología de desarrollo para la Actividad productiva de la UCI*. . Hbana: UCI.
46. SMYTH, N., 2017. *Android Studio 2.3 Development Essentials - Android 7 Edition*. S.I.: PayloadMedia, Inc.
47. SOLÍS, I.S. y TINOCO, E.E.C., 2015. *COMPENDIO DE INGENIERÍA DE SOFTWARE*. S.I.: Iván Soria Solís. ISBN 978-612-4281-04-4.
48. SOMMERVILLE, I., 2011a. *Ingeniería del software*. S.I.: Pearson Educación. ISBN 978-84-7829-074-1.
49. SOMMERVILLE, I., 2011b. *Ingeniería del software*. NOVENA EDICIÓN. S.I.: Pearson Educación. ISBN 978-607-32-0603-7.
50. SORIANO, J.E.A., 2011. *Android: Programación de dispositivos móviles a través de ejemplos*. S.I.: Marcombo. ISBN 978-84-267-1767-2.
51. SOSA, L.A.M., RUMBAUT, G.I.P., ESTRADA, G.J., MARGARITA, A., REY, L.R. del, LOZANO, A.S. y SERRANO, S.C.W., 2012. Satisfacción de estudiantes y profesores con el taller de integración en Morfofisiología Humana III. *Revista Electrónica de las Ciencias Médicas en Cienfuegos* [en línea], vol. 10, no. 6, pp. 6. [Consulta: 28 septiembre 2016]. DOI 1727-897X. Disponible en: <http://www.medisur.sld.cu/index.php/medisur/article/view/2265/7472>.
52. SOTELO, A.F. y ARVELO, M.V., 2015. Proceso de Evaluación del Comportamiento Estudiantil para Potenciar su Desempeño. *CIAIQ2015* [en línea], vol. 2, no. 0. [Consulta: 28 septiembre 2016]. Disponible en: <http://proceedings.ciaiq.org/index.php/ciaiq2015/article/view/261>.
53. UNAM, F. de Q. de la, 2016. *Farmacovigilancia Mobile App, construir una aplicación usando Constructor Appy Pie App Gratis*. [en línea]. [Consulta: 1 mayo 2017]. Disponible en: <http://essnappy.appypie.com/marketplace/iphone-android/farmacovigilancia>.

Anexos

Anexo: 1 Encuesta aplicada a los médicos para medir sus conocimientos con respecto a los medicamentos que más RAM causaron en el año 2015

Años de experiencia: _____ Especialidad: _____
 Tipo de US: _____

Esta encuesta tiene como objetivo recoger su opinión sobre si es necesario realizar una herramienta informática, para dispositivos móviles, que les permita acceder a reportes de las reacciones adversas a medicamentos (RAM) ocurridas en Cuba.

1. ¿Tiene Ud. dispositivos móviles? Marque con una X su respuesta.

Teléfono inteligente ___ Tablet ___ Ninguno ___

2. ¿Qué tipo de sistema operativo tiene su dispositivo móvil? Marque con una X su respuesta

Android ___ IOS ___ Windows Phone ___ Otros ___

3. Si tuviera una herramienta que le permitiera: Marque con una X su respuesta

Crear reportes de una base de datos de más de 5 años de información históricas de RAM ocurridas en Cuba y Ud. Pudiera realizar consultas a dicha base de datos para hacer análisis de (medicamentos, grupo etario, antecedentes patológico y sexo)

¿La usaría?: Sí ___ No ___ No sé ___

4. ¿Tiene Ud. conocimiento de algún sistema informático cubano que le permita realizar la actividad antes planteada? Marque con una X su respuesta.

Sí ___ No ___

5. Si Ud. tuviera en su dispositivo móvil dicha herramienta, ¿La usaría? Marque con una X su respuesta.

Sí ___ No ___ No sé ___

6. ¿Conoce UD. cual de los siguientes medicamentos fue el que más reacciones adversas provocó en el año 2015? Marque con una X.

Penicilina Rapilenta ___ Ibuprofen ___ Vacuna Pentavalente ___
 No sé ___

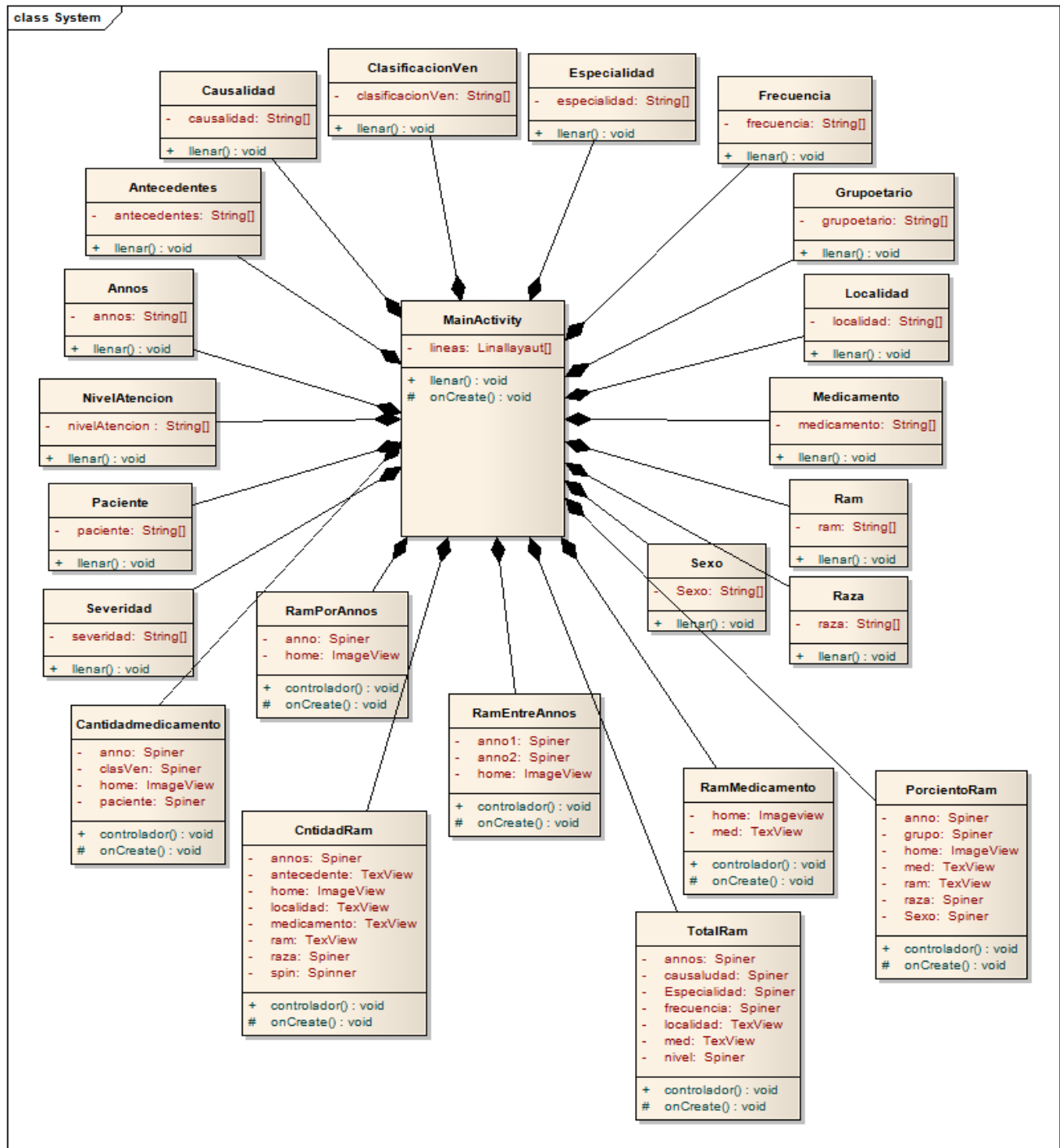
7. ¿Cree UD. necesario conocer las cantidad de RAM ocurridas en Cuba? Marque con una X su respuesta.

Sí ___ No ___ No sé ___

Anexo 2: Modelo 33-36-02 de recogida de RAM del MINSAP

Modelo 33-36-02 MINISTERIO DE SALUD PÚBLICA Farmacia	NOTIFICACIÓN DE SOSPECHA DE REACCIÓN ADVERSA A MEDICAMENTOS POR EL PROFESIONAL SANITARIO										Pág. 1 de 1				
											Mes				
											Año				
Provincia				Municipio				Barrio							
PACIENTE: Nombre y apellidos _____ Edad: _____ Sexo: F <input type="checkbox"/> M <input type="checkbox"/> Color de la Piel: B <input type="checkbox"/> N <input type="checkbox"/> A <input type="checkbox"/> M <input type="checkbox"/> Peso (kg) _____															
Nombre del que reporta				Médico <input type="checkbox"/>		Lic. Farmacia <input type="checkbox"/>		Téc. Farmacia <input type="checkbox"/>		Lic. Enfermería <input type="checkbox"/>					
				Enfermera <input type="checkbox"/>		Estomatólogo <input type="checkbox"/>		Otro <input type="checkbox"/>							
Medicamento tomado hasta 3 meses antes de la RAM. Marcar con una cruz los sospechosos	Lote	Via de administración	Dosis diaria	TRATAMIENTO						Motivo de Prescripción					
				Inicio			Fin								
				Día	Mes	Año	Día	Mes	Año						
Fabricante:				Patologías que presenta el paciente											
REACCIONES. Enumérense por separado				Fecha de Inicio			Fecha de Término			Requirió ingreso o prolongó su estadia hospitalaria		Atención de Urgencia		Puso en peligro su vida	
				Día	Mes	Año	Día	Mes	Año	SI	NO	SI	NO	SI	NO
DESENLACE															
Recuperado				¿ Se ha suspendido la medicación? SI NO											
No recuperado				¿ Ha mejorado al suspenderla? SI NO											
Recuperado con secuelas				¿ Se administró nuevamente este medicamento? SI NO											
Mortal				¿ Si se administró nuevamente hubo recurrencia de síntomas? SI NO											
Observaciones adicionales				Fecha de notificación Día Mes Año											

Anexo 3: Diagrama de clases



Anexo 4: Caso Prueba caso uso porciento RAM.

SC Porciento de RAM								
Escenario	Descripción	RAM	SEXO	Raza	Fecha inicio	Fecha fin	Grupo etario	Medicamento
EC 1.1 Generar reporte porciento de RAM	Se muestra correctamente el reporte	V	V	V	V	V	V	V
		Fiebre	F	Blanca	2013	2015	niño	Ibuprofeno
		V	V	V	V	I	V	V
		Vómito	M	Negra	2014	2013	adulto	Vac Pentavalente
EC 1.2: Datos incorrectos	Se introdujeron datos inválidos	I	V	V	V	V	V	V
		Dientes manchados	F	Mestiza	2015	2015	gediátrico	ciprofoxacina
		V	V	V	V	V	I	V
		Dolor estómago	M	Blanca	2011	2014	jóven	Penicilina Rap

Anexo 5: Encuesta de satisfacción aplicada a los médicos

Años de experiencia: _____ Especialidad: _____
 Tipo de US: _____

Esta encuesta tiene como objetivo obtener su opinión sobre la aplicación móvil Android RamCuba, esta aplicación proporciona reportes de reacciones adversas a medicamentos (RAM) ocurridas en Cuba en un periodo de tiempo determinado.

1. ¿Considera Ud. importante utilizar esta aplicación?

Marque con una X su respuesta.

Sí ___ No ___ No sé ___

2. ¿La aplicación RamCuba soluciona los problemas existentes del poco conocimiento de las RAM ocurridas en Cuba?

Marque con una X su respuesta

Totalmente de acuerdo ___ Totalmente en desacuerdo ___

De acuerdo ___ Ni de acuerdo ni en desacuerdo ___ Desacuerdo ___

3. Si tuviera la aplicación RamCuba ¿La usaría?

Marque con una X su respuesta

Sí ___ No ___ No sé ___

4. ¿Le satisface los resultados de los reportes de RAM, (tanto gráficos como nominales) que muestra RamCuba?

Marque con una X su respuesta.

Me satisface mucho ___ No me satisface tanto ___ Me da lo mismo ___

No me satisface nada ___ Me insatisface más de lo que me satisface ___

No se que decir ___

5. ¿Cree Ud. que conocer la información de los reportes de RAM a través de la aplicación RamCuba, le ayuda a evitar reacciones en el momento de prescribir un medicamento?

Marque con una X su respuesta.

Totalmente de acuerdo ___ Totalmente en desacuerdo ___

De acuerdo ___ Ni de acuerdo ni en desacuerdo ___ Desacuerdo ___

6. ¿Cree Ud. que conocer a través de la aplicación RamCuba el municipio donde ocurrieron más RAM, en un tiempo determinado, le permite realizar estudios demográficos para tomar medidas a tiempo?

Marque con una X su respuesta.

Totalmente de acuerdo ___ Totalmente en desacuerdo ___

De acuerdo ___ Ni de acuerdo ni en desacuerdo ___ Desacuerdo ___

7. ¿La aplicación RamCuba le permitió conocer la cantidad de RAM ocurridas en Cuba en un periodo de tiempo siendo esto de gran importancia para los médicos y farmacoepidemiólogos?

Marque con una X su respuesta.

Totalmente de acuerdo ___ Totalmente en desacuerdo ___

De acuerdo ___ Ni de acuerdo ni en desacuerdo ___ Desacuerdo ___