



Universidad de las Ciencias Informáticas
Facultad 2

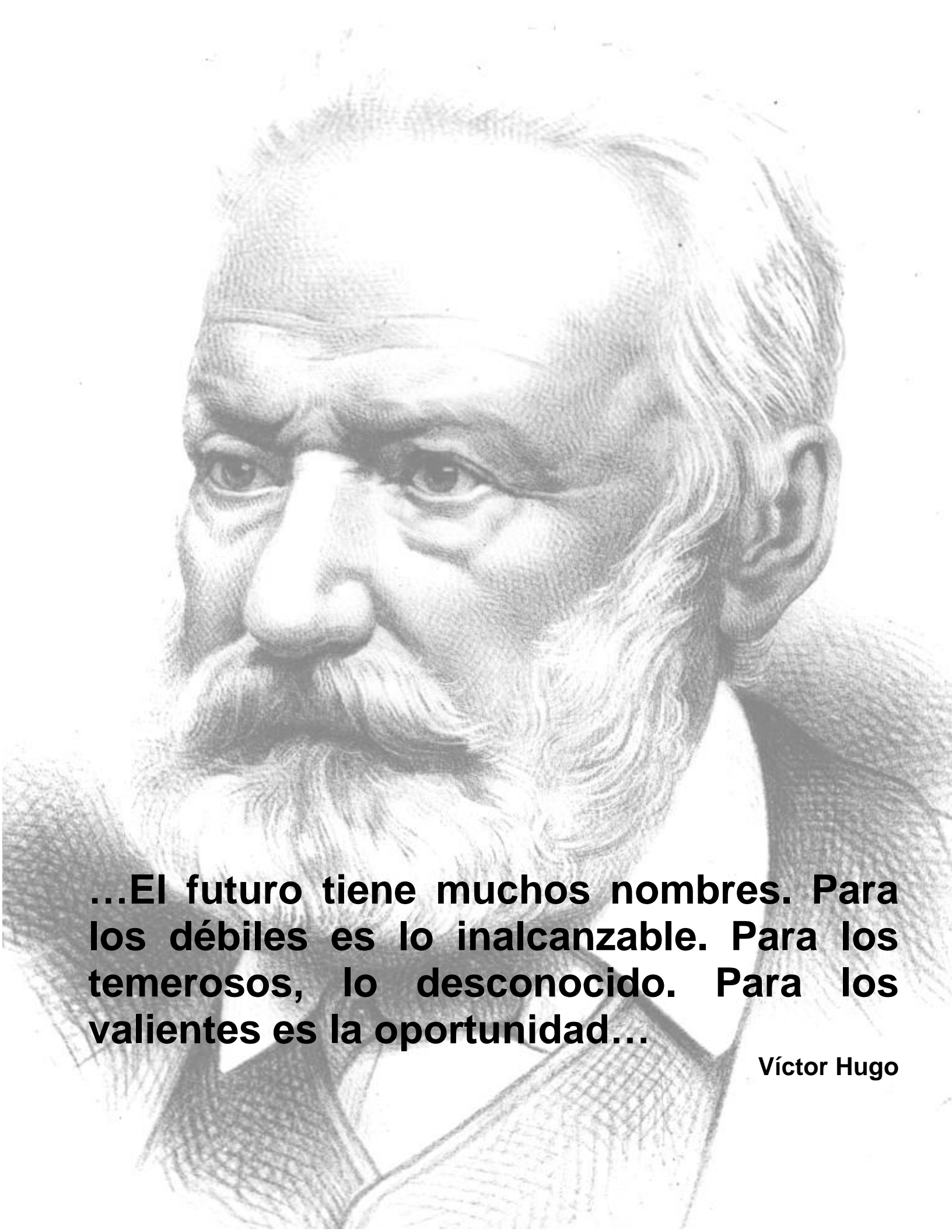
Tema: Sistema Cliente-Servidor SIIPOL MÓVIL.

**Trabajo de Diploma para optar por el título de
Ingeniero Informático**

Autor: ReynierMoreuMorell.
Carlos León Castellanos

Tutor: Ing. Damián Ilizastegui Arriba.
Ing. Lex Karel Zayas Hernández

Junio 2013



...El futuro tiene muchos nombres. Para los débiles es lo inalcanzable. Para los temerosos, lo desconocido. Para los valientes es la oportunidad...

Víctor Hugo

DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 2 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Reynier Moreu Morell

Carlos León Castellanos

Ing. Damián Ilizastegui Arriba.

Firma de Autor

Firma de Autor

Firma de Tutor

Datos de Contactos

Ing. Damián Ilizastegui Arriba: Graduado en el 2007 de Ingeniero en Ciencias Informáticas. Experiencia de 6 años en el desarrollo de aplicaciones empresariales y de telecomunicaciones. Arquitecto del proyecto SIIPOL MÓVIL.

AGRADECIMIENTOS DE:

Reynier Moreu Morell

- *A mi madre, por ese amor incondicional, por hacerme sentir su protección, su presencia y cariño en todo momento.*
- *A mi padre, por ser para mí ese ejemplo de esfuerzo, constancia y fuerza que deseo seguir, por darme siempre la seguridad de su apoyo y respaldo.*
- *A Elieta, mi amiga, mi confidente, mi conciencia. Por estar siempre allí, por soportar mi carácter, mis momentos de estrés y crisis. (Gracias Ppu, siempre serás mR).*
- *A mis tutores, por no ser insoportables como algunos, pero estar allí cuando los necesite, y en especial al profe Damián por su ayuda, la cual no se puede medir ni contar.*
- *A Andrés Selles, por ser un amigo incondicional y ayudarme cuando lo necesite, así como demostrarme que hay personas noble y de buen corazón.*
- *A Alejandro Expósito por toda su ayuda, por sus palabras duras pero ciertas, por abrirme los ojos con la verdad y por enseñarme que no basta saber, sino saber aún mucho más.*
- *A mi profe Josefina, alguien a quien le debo estar donde estoy y nunca lo olvidaré.*
- *A los amigos que no están a mi lado, a los que están, a los que dijeron cosas buenas y malas, a los que aportaron y a los que no. A mi familia, por formar mi carácter y enseñarme lo que se debe valorar y lo que no. Quiero darle gracias a todos, porque gracias a ustedes soy quien soy.*
- *Pero hay un grupo de nombres que no pueden faltar y quiero me perdonen los que no sean mencionados: “Yusmel (hermano), Omar (hermano), Lidiagnis (la tutu y hermanita), Iliana (hermana), Felipe (mi hermano pinareño)” A los hermanos del paso de escalera y apartamento (Robe, Andy, Gabriel el Pro, Yankiel) a los viciosos del HON (Olivio el matarife, Arian, al tocayo, Pedro) y todos los que saben que los estimo y recordaré siempre.*

AGRADECIMIENTOS DE:

Carlos León Castellanos

Le agradezco a Damián, mi tutor, por su apoyo incondicional.

También a mi tío Omar por su paciencia y dedicación.

Los dos fueron decisivos en esta etapa final, gracias a los dos; amigos.

Los tres vamos a triunfar.

DEDICATORIA

DEDICATORIA DE:

Reynier Moreu Morell

*A mi madre y mi padre,
Por ser mi fuerza para seguir adelante.
Todo de mi estará siempre dedicado a ustedes.*

DEDICATORIA DE:

Carlos León Castellanos

Dedico mi esfuerzo a quienes me ayudan a vivir, en especial a mis padres por haberme dado la vida, a mi hermano por su eterna compañía, a mi familia y amigos que me hacen sonreír.

RESUMEN

En Venezuela, el sistema SIIPOL MOVIL WAP es una herramienta importante para la lucha contra el crimen, esta permite consultar información operativa a través de un móvil, permitiendo que los oficiales de policía puedan actuar de manera rápida ante cualquier situación.

El sistema SIIPOL MOVIL WAP es accedido a través del navegador del móvil, por lo cual mucha de la información que viaja desde el servidor al navegador es código XHTML, que en gran parte solo influye en la manera que se muestra la información; esto introduce demoras por la cantidad de texto que se transmite. También algunas de las tecnologías que fueron utilizadas en su desarrollo están quedando obsoletas y el sistema carece de una arquitectura bien pensada; lo que da paso al desarrollo del sistema cliente-servidor SIIPOL Móvil. Este es accedido por una aplicación en el móvil que sabe de antemano como mostrar los datos, la única información intercambiada con el servidor son los parámetros y los resultados de las búsquedas.

En el documento se encuentran las principales características de otros sistemas de gestión de información policial, que se consultaron como parte de la investigación. Se recogen las principales características de las herramientas empleadas para la implementación de la solución y se tratan las principales características de la metodología empleada.

Palabras claves: Servicio web REST, Android, Sistema Cliente-Servidor.

INTRODUCCIÓN	15
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	18
1.1 INTRODUCCIÓN	18
1.2 MARCO CONCEPTUAL	18
1.2.1 SERVICIOS WEB	18
CARACTERÍSTICAS DE LOS SERVICIOS WEB	18
1.2.2 SERVICIOS WEB REST	19
CARACTERÍSTICAS	19
PRINCIPIOS FUNDAMENTALES	20
1.3 SISTEMAS SIMILARES	21
1.4 METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	22
1.4.1 PROGRAMACIÓN EXTREMA (EXTREME PROGRAMMING, XP)	22
1.5 LENGUAJE DE PROGRAMACIÓN	23
1.5.1 MOTIVOS PARA USARLO COMO LENGUAJE DE PROGRAMACIÓN:	23
1.6 HERRAMIENTAS DE DESARROLLO	23
1.6.1 ENTORNO DE DESARROLLO INTEGRADO	23
1.6.2 ECLIPSE	24
1.7 SISTEMAS OPERATIVOS PARA MÓVILES	24
1.8 PLATAFORMA DE DESARROLLO	25
1.9 FRAMEWORK.....	26
1.9.1 HIBERNATE VERSIÓN 3.....	26
1.9.2 SPRING VERSIÓN 3.1.2	26
CONCLUSIONES PARCIALES.....	27
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN	28
2.1 INTRODUCCIÓN	28
2.2 PROPUESTA DE SOLUCIÓN.....	28
2.3 REQUERIMIENTOS DEL SISTEMA	29
2.3.1 REQUISITOS NO FUNCIONALES	29
REQUISITOS NO FUNCIONALES DE LA APLICACIÓN CLIENTE	29
REQUISITOS NO FUNCIONALES DE LA APLICACIÓN SERVIDOR	30
2.4 EXPLORACIÓN.....	31
HISTORIAS DE USUARIO (HU).....	31

2.5 PLANIFICACIÓN.....	32
ESTIMACIÓN DE ESFUERZO POR HU.	32
2.6 PLAN DE ITERACIONES.....	33
2.7 PLAN DE DURACIÓN DE ITERACIONES.	34
2.8 PLAN DE ENTREGAS.....	35
CONCLUSIONES PARCIALES.....	35
CAPÍTULO 3: DISEÑO	36
3.1 INTRODUCCIÓN	36
3.2 ARQUITECTURA DE LA SOLUCIÓN.....	36
3.3 PATRÓN ARQUITECTÓNICO N CAPAS	36
3.3.1 PRINCIPIOS CLAVES.....	36
3.3.2 BENEFICIOS	36
3.4 PATRONES DE DISEÑO.	37
3.4.1 PATRONES PARA ASIGNAR RESPONSABILIDAD.	37
3.4.2 PATRONES GoF	38
3.5 TARJETAS CLASE-RESPONSABILIDAD-COLABORACIÓN (CRC).....	38
3.5.1 TARJETAS CRC PARA EL CLIENTE.....	39
3.5.2 TARJETAS CRC PARA EL SERVIDOR.....	39
CONCLUSIONES PARCIALES.....	39
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA	40
4.1 INTRODUCCIÓN	40
4.2 TAREAS DE INGENIERÍA.....	40
4.3 VALIDACIÓN DE LA PROPUESTA	40
4.3.1 LAS PRUEBAS DE ACEPTACIÓN	41
CONCLUSIONES PARCIALES.....	42
CONCLUSIONES GENERALES	43
RECOMENDACIONES	44
REFERENCIAS BIBLIOGRÁFICAS	45
ANEXOS.....	47
ANEXO 1: HISTORIAS DE USUARIOS.....	47
ANEXO 2: TAREAS DE INGENIERÍAS.....	55
ANEXO 3: TARJETAS CRC	60

ÍNDICE DE TABLAS

Tabla 1. Descripción de las personas relacionadas con el Sistema	31
Tabla 2.Registrar Traza de Consulta Sobre Entidad Operativa.	32
Tabla 3. Estimación de esfuerzo por HU	33
Tabla 4. Plan de Duración de Iteraciones.....	35
Tabla 5. Plan de entregas	35
Tabla 6. Tarea de Ingeniería: Consultar Personas con Registros Policiales.....	40
Tabla 7: Resultados de las Pruebas de Aceptación realizadas en la primera Iteración	41
Tabla 8: Resultados de las Pruebas de Aceptación realizadas en la segunda Iteración.....	42

INTRODUCCIÓN

La seguridad en la sociedad, es un aspecto obligatorio a la hora de medir la calidad de vida de sus habitantes, siendo una preocupación constante para los gobiernos y los mandatarios locales, que ven como los ciudadanos reclaman y necesitan vivir con tranquilidad, para no ser víctimas de la delincuencia.

América Latina muestra en los últimos años la más notable baja en el nivel de paz a causa de más violencia interna, homicidios y mayores niveles de criminalidad percibida. La nación venezolana es uno de los países más violentos de esta región (1).

Venezuela cuenta con varios cuerpos de investigación contra el crimen y la delincuencia, entre los que se encuentra el Cuerpo de Investigaciones Científicas Penales y Criminalísticas (CICPC). Con el objetivo de incrementar la eficiencia en el sistema de respuestas del CICPC, se creó el producto SIIPOL Móvil Wap, que le permite a los funcionarios, acceder a la base de datos del Sistema de Investigación e Información Policial (SIIPOL) mediante dispositivos móviles y una aplicación web. Este sistema brinda mayor eficiencia en las respuestas operativas, ya que los usuarios autorizados pueden realizar consultas sobre entidades operativas (armas, vehículos, objetos y personas), fundamentalmente cuando estos se encuentren fuera de la institución.

SIIPOL-Móvil Wap es una aplicación Web que es accedida por los navegadores de los teléfonos móviles. Fue creada con tecnologías que están quedando obsoletas actualmente, las cuales son el framework Spring 2.5 el cual carece de funcionalidades que poseen sus versiones posteriores, la librería de conversión de imágenes Alembik, que es usada en la aplicación y era software libre, cambió su licencia a una privativa, eliminando la posibilidad de obtener soporte para la misma y el framework Hibernate que también será utilizado en su última versión; este problema, sumado a otros de diseño y arquitectura tales como la repetición de código; la arquitectura no fuese pensada para la adicción de más funcionalidades y la poca portabilidad entre los módulos le restan flexibilidad. Además, la aplicación presenta problemas de rendimiento debido al tráfico innecesario de contenido HTML al ser accedido en los navegadores de los dispositivos móviles, lo cual dificulta el trabajo de los funcionarios cuando se encuentran en zonas de baja conectividad.

A partir de lo anteriormente expuesto se plantea cómo **problema a resolver**: ¿Cómo resolver los problemas de rendimiento en el acceso a la información operativa del CICPC?

Este problema se enmarca en el **objeto de estudio** sistemas de gestión de la información policial y el **campo de acción** sistemas para dispositivos móviles para la gestión de la información policial del CICPC.

El **objetivo general** es la implementación de un sistema cliente-servidor para móviles que consulte la información operativa del CICPC.

Se plantean los **objetivos específicos**:

1. Desarrollar una capa de servicios REST para la búsqueda de información operativa en las bases de datos del CICPC.
2. Desarrollar un subsistema para la administración de usuarios y permisos de la aplicación.
3. La implementación de un cliente para móviles Android, que permita consultar la información operativa del CICPC.

Se espera implementar un sistema, que consultará la información operativa del CICPC a través de servicio REST, que resuelva los problemas de rendimiento que presenta la versión anterior.

Con el propósito de guiar, controlar, evaluar y perfilar el trabajo hacia el alcance de los objetivos trazados, se definieron las siguientes **tareas**:

1. Análisis de las actuales tendencias en los sistemas de gestión policial.
2. Elaboración del estado del arte de software para el acceso a la información policial.
3. Selección de las herramientas, metodología de desarrollo de software, lenguaje de programación y frameworks a utilizar en la implementación del sistema.
4. Análisis de la estructura y lógica de la base de datos de CICPC haciendo énfasis en la base de datos de SIIPOL MÓVIL.
5. Definición de las interfaces visuales de la aplicación cliente para móviles Android.
6. Definición de la estructura, parámetros y dirección de los servicios REST.
7. Implementación de la lógica de negocio de la aplicación para dispositivos Android.
8. Implementación en Android de una capa para el consumo de servicios REST.
9. Definición e implementación de un modelo de autenticación basado en roles y permisos.
10. Implementación de la lógica de negocio de la aplicación servidor.
11. Implementación de una capa de acceso a la base de datos del CICPC.

Entre los métodos de investigación científica existentes se utilizaron los siguientes:

Métodos Teóricos:

Analítico-Sintético: Con el objetivo de analizar documentos, trabajos de diplomas, sitios web donde se muestre información referente a los Sistemas de apoyo a la gestión de servicios policiales. Permitiendo la recopilación de los elementos más importantes para el desarrollo del presente trabajo de investigación.

Métodos Empíricos:

Observación: Utilizado para tener una visión general de los requerimientos del sistema, las posibles restricciones y propiedades, así como para determinar las posibles soluciones durante todo el proceso de desarrollo del software.

El presente trabajo consta de una introducción, cuatro capítulos, conclusiones generales, recomendaciones, las referencias bibliográficas utilizadas en el desarrollo del trabajo y los anexos.

Capítulo 1: Fundamentación teórica: Aborda de forma general los aspectos teóricos de la investigación realizada, el estado del arte y las herramientas seleccionadas, concluyendo con una propuesta de solución para la implementación de la aplicación cliente para móvil SIIPOL Móvil.

Capítulo 2: Propuesta de solución: Características del Sistema. Exploración y Planificación. Se define lo que debe hacer el sistema a partir de las funcionalidades requeridas y las restricciones impuestas por el cliente.

Capítulo 3: Diseño: Se describe el proceso de implementación del sistema y la descripción de las clases del diseño utilizando las tarjetas Clase-Responsabilidad-Colaboración (CRC).

Capítulo 4: Implementación y Prueba: Se describe todo el proceso de implementación del sistema. Se realizan los casos de prueba al sistema implementado y se muestran los resultados de cada una de las iteraciones de pruebas realizadas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se analizan conceptos asociados a la investigación, tecnologías y metodologías utilizadas para la implementación del producto Siipol Móvil, así como el estado del arte referente a las tendencias actuales de estas herramientas.

1.2 Marco conceptual

1.2.1 Servicios Web

Los servicios web surgen como consecuencia de la necesidad de crear un estándar para la comunicación entre las distintas plataformas y lenguajes de programación existentes.

Existen numerosas definiciones sobre lo que son los Servicios Web, lo que muestra su complejidad a la hora de dar una adecuada definición que englobe todo lo que son e implican. Una definición general y convincente es que un Servicio Web es un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web (2).

Se puede decir, que un servicio web es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones, sin importar que estén desarrolladas en lenguajes de programación distintos y que se ejecuten sobre plataformas diferentes.

Características de los servicios web

Proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Proporcionan interoperabilidad y extensibilidad entre estas aplicaciones (2).

- Permiten exponer y hacer disponibles funcionalidades de los sistemas informáticos mediante tecnologías y protocolos WEB estándar.
- Cada servicio web se responsabiliza en realizar un conjunto de funciones concretas y bien definidas.
- Los servicios web actúan como componentes independientes que se pueden integrar para formar sistemas distribuidos complejos.

1.2.2 Servicios Web REST

La transferencia de estado representacional (REST) es un término que se originó en el año 2000 en una tesis de doctorado sobre la Web escrita por Roy Fielding siendo este uno de los autores principales de la especificación HTTP y una autoridad citada frecuentemente en la materia de Arquitectura de Redes.

REST es una abstracción de los elementos arquitectónicos dentro de un sistema hipermedia distribuido. Ignora los detalles de la implementación de los componentes y la sintaxis del protocolo con el fin de centrarse en las funciones de los componentes, las restricciones sobre su interacción con otros componentes, y su interpretación de los elementos de datos significativos. Abarca las limitaciones fundamentales sobre los componentes, conectores, y datos que definen la base de la arquitectura de la Web, y por lo tanto la esencia de su comportamiento como una aplicación basada en la red (3).

REST define un grupo de principios arquitectónicos por los cuales se diseñan servicios web haciendo foco en los recursos del sistema, incluyendo cómo se accede al estado de dichos recursos y cómo se transfieren por HTTP hacia clientes escritos en diversos lenguajes. REST emergió en los últimos años como el modelo predominante para el diseño de servicios. De hecho, logró un impacto tan grande en la web que prácticamente logró desplazar a SOAP y las interfaces basadas en WSDL por tener un estilo bastante más simple de usar.

Características

- En arquitecturas REST, los servicios no publican un conjunto arbitrario de métodos u operaciones.
- En REST lo que se publica son *recursos*. Un recurso se puede considerar como una entidad que representa un concepto de negocio que puede ser accedido públicamente.
- Cada recurso, como buena entidad que se precie, y de acuerdo a los principios Orientado a Objeto, posee un identificador único y global, que lo distingue de cualquier otro recurso, aunque ambos tuvieran exactamente los mismos datos.
- Cada recurso posee un estado interno, que no puede ser accedido directamente desde el exterior. Lo que sí es accesible desde el exterior es una o varias representaciones de dicho estado. Por representación se entiende un formato de datos concreto usado para la transferencia de una copia del estado público del recurso entre el cliente y el servidor. La implementación del recurso decide qué información es visible o no desde el exterior, y que representaciones de dicho estado se soportan.

- En REST todos los recursos comparten una interfaz única y constante. Todos los recursos tienen las mismas operaciones. Las operaciones nos permiten manipular el estado público del recurso. En un sistema REST típico se definen cuatro operaciones.
 - ✓ **CREATE.** En esta operación el cliente manda al servidor una petición para crear un nuevo recurso. Opcionalmente el cliente puede mandar una representación del estado inicial de este recurso. El servidor responde con el identificador global del nuevo recurso.
 - ✓ **DELETE.** En esta operación el cliente elimina un recurso del servidor. El cliente necesita saber el identificador del recurso.
 - ✓ **READ.** Con esta operación el cliente puede leer una representación del estado de un recurso, identificado con su identificador global. El cliente puede especificar qué tipos de representaciones entiende. Aquí lo que ocurre realmente es que se copia el estado del recurso en el servidor y se pega en el cliente. Ambas copias del estado no se mantiene sincronizadas. El servidor puede cambiar el estado real del recurso y el cliente, de forma independiente, puede modificar su copia local del estado del recurso.
 - ✓ **UPDATE.** Como el servidor y el cliente tienen una copia diferente del estado, el cliente puede usar esta operación para sobrescribir o grabar su copia del estado en el servidor. De esta manera se puede actualizar el estado del recurso con las modificaciones hechas en el cliente.
- La implementación del servicio es libre de prohibir alguno de estos métodos para un recurso en concreto. También debe definir el modelo de datos que se va a publicar y que representaciones soporta. Lo que no puede hacer es inventarse operaciones de forma arbitraria. Las operaciones son siempre las mismas.
- Los distintos recursos se pueden interrelacionar y referenciar entre sí mediante sus identificadores globales (4).

Principios fundamentales

- REST utiliza los métodos HTTP de manera explícita:

REST hace que los desarrolladores usen los métodos HTTP explícitamente de manera que resulte consistente con la definición del protocolo. Este principio de diseño básico establece una asociación uno-a-uno entre las operaciones de crear, leer, actualizar y borrar y los métodos HTTP. De acuerdo a esta asociación:

- ✓ se usa POST para crear un recurso en el servidor
- ✓ se usa GET para obtener un recurso
- ✓ se usa PUT para cambiar el estado de un recurso o actualizarlo
- ✓ se usa DELETE para eliminar un recurso.

- REST expone URIs con forma de directorios:

Desde el punto de vista de un cliente que accede a un recurso, la URI determina qué tan intuitivo va a ser el servicio REST y si el servicio va a ser utilizado tal como fue pensado al momento de diseñarlo. Las URI de los servicios web REST deben ser intuitivas, hasta el punto de que sea fácil adivinarlas. Las URI son como una interfaz auto-documentada que necesita de muy poca o ninguna explicación o referencia para que un desarrollador pueda comprender a lo que apunta y a los recursos derivados relacionados. Una forma de lograr este nivel de usabilidad es definir URIs con una estructura al estilo de los directorios. Este tipo de URIs es jerárquica, con una única ruta raíz y va abriendo ramas a través de las sub-rutas para exponer las áreas principales del servicio. Por tanto, una URI no es solamente una cadena de caracteres delimitada por barras, sino más bien un árbol con subordinados y padres organizados como nodos.

- REST no tiene estado:

Los **servicios sin estado** son mucho más simples de diseñar, escribir y distribuir a través de múltiples servidores. Un servicio sin estado no sólo funciona mejor, sino que además mueve la responsabilidad de mantener el estado al cliente de la aplicación. En un servicio web REST, el servidor es responsable de generar las respuestas y proveer una interfaz que le permita al cliente mantener el estado de la aplicación por su cuenta. Este aspecto del diseño de un servicio web REST puede descomponerse en dos conjuntos de responsabilidades, como una separación de alto nivel que define cómo puede mantenerse un servicio sin estado (5).

1.3 Sistemas similares

En la literatura consultada se identificaron varios trabajos que dan soluciones a los Sistemas de apoyo a la gestión de servicios policiales. Algunos de los ejemplos más significativos son:

- **WM Police**¹: es una solución móvil que opera sobre un dispositivo portátil y permite a los agentes de tránsito imponer comparendos haciendo consultas en línea sobre antecedentes, tarifas vigentes, legislación, estado de carteras del usuario, proceso de contravenciones y otros elementos. De la misma manera puede hacer el recaudo inmediato a través de tarjetas de Crédito, disminuyendo la cartera al facilitar su recaudo (6).
- **FALCON**²: es el sistema de última generación para identificar a sospechosos de crímenes y para reportar datos. Los usuarios autorizados pueden identificar biométricamente a sospechosos y extraer información desde bases de datos para criminales desde fuentes estatales y nacionales (7).

¹Herramienta utilizada por las Secretarías de Tránsito en Colombia para mejorar el recaudo por concepto de las multas y comparendos

²Herramienta utilizada por La Florida (EEUU) para la identificación de criminales.

- **Operaciones Móviles de Policía³**: aplicación móvil para Smartphone BlackBerry, desarrollada por Tecnew Informática. La aplicación permite a los agentes de policía acceder a bases de datos y a los registros policiales desde sus Smartphone BlackBerry, tales como información actualizada sobre personas, vehículos, conductores y antecedentes criminales (8).

Los sistemas anteriormente mencionados, dan soluciones a problemas similares que son resueltos con la implementación del producto SIIPOL Móvil, pero no son factibles debido a su carácter privativo y su alto costo de despliegue, ya que sus dispositivos móviles poseen características específicas de fabricación, mientras que el producto SIIPOL Móvil, será soportado por todos los teléfonos que soporten las aplicaciones Android versión 2.2 o superior, siendo esto una ventaja, ya que los productos de Android dominan el mercado actual y son de los teléfonos más vendidos y utilizados a nivel internacional(33).

1.4 Metodología de desarrollo de software

La dirección del proyecto SIIPOL Móvil, decidió establecer las herramientas, metodología y lenguaje de desarrollo a utilizar, por tal motivo no forma parte del presente trabajo investigar acerca de éstas, sino centrarse en el estudio de la metodología previamente seleccionada.

1.4.1 Programación extrema (extreme programming, XP)

XP es una metodología que sigue la filosofía de las metodologías ágiles, cuyo objetivo es conseguir la máxima satisfacción del cliente de forma rápida y eficiente ante los cambios de requisitos. XP (9) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en intercambio continuo entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Propone realizar diseños simples, códigos simples y proporcionar rápida respuesta de lo requerido y lograr un cliente contento.

Se sustituye la documentación escrita por la comunicación directa entre clientes y desarrolladores o entre los propios desarrolladores. Propone un desarrollo iterativo a través de cuatro pasos, planificación, diseño, codificación y prueba. En cada iteración se añaden nuevas funcionalidades al software. Es especialmente adecuada para proyectos con requisitos imprecisos y muy variables además donde existe un alto riesgo técnico.

³Herramienta utilizada por Brasil con el objetivo de combatir la delincuencia de forma eficiente y proteger a la comunidad.

Se basa en una serie de prácticas y principios que se han ido gestando a lo largo de toda la historia de la ingeniería del software, son de sentido común pero llevadas al extremo. Usadas conjuntamente proporcionan una nueva metodología de desarrollo software que se puede englobar dentro de las metodologías ágiles o ligeras (10).

Las razones fundamentales que se tomaron en cuenta para escoger esta metodología:

- El proyecto es pequeño: XP está concebida para ser utilizada dentro de proyectos pequeños y de desarrollo rápido, se adapta perfectamente a este caso en que la duración será de aproximadamente 6 meses.
- Pocos roles: esta metodología está dirigida a grupos de desarrollo pequeños y con pocos roles, en este caso el equipo está compuesto por 2 integrantes.
- El manejo de los cambios: a medida que el proyecto avanza pueden surgir nuevas expectativas o ideas que pueden ser incorporadas fácilmente permitiéndole mayor adaptabilidad al producto, con la metodología XP esto es completamente factible pues esta se adapta perfectamente a los proyectos cuyos requerimientos cambian a menudo.

1.5 Lenguaje de programación

Java: es un lenguaje de programación de propósito general. En la actualidad su uso está muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Es creado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras (11).

1.5.1 Motivos para usarlo como lenguaje de programación:

El principal motivo para la utilización de java como lenguaje de programación es que Android (*Descrito más adelante*) tiene Java como lenguaje base. A eso se suma que es un lenguaje multiplataforma, lo que permite su ejecución en diferentes sistemas operativos y que está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos.

1.6 Herramientas de desarrollo

1.6.1 Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (IDE) es un conjunto de herramientas utilizadas por los programadores, que incluye por lo general, un buen editor de código, administrador de proyectos y archivos, enlace a compiladores e integración con sistemas controladores de versiones o repositorios, además de brindar facilidades para la construcción de interfaces gráficas de usuario (12).

Las características principales de un Entorno de Desarrollo Integrado son las siguientes:

- Permite trabajar con varios proyectos a la vez, encontrar código duplicado y personalizar el entorno.
- Incorpora completamiento de código.
- Permite la integración con aplicaciones de control de versiones.
- Permite comparar archivos.
- Mantiene un historial local de los archivos.
- Incorpora funcionalidades a través de plugins⁴.

1.6.2 Eclipse

Eclipse es un entorno de desarrollo integrado (IDE, Integrated Development Environment) que facilita enormemente las tareas de edición, compilación y ejecución de programas durante su fase de desarrollo. Aunque Eclipse pretende ser un entorno versátil soportando varios lenguajes de programación, es con el lenguaje Java con el que mejor se integra y con el que ha ganado su popularidad. Debido a su fácil integración con Java, se decide utilizarlo como IDE de desarrollo (12).

Se utilizó para el desarrollo de la aplicación Servidor y el Cliente.

1.7 Sistemas operativos para móviles

Alguno de los sistemas operativos para móviles más usados son:

- **Android:** Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS, Symbian y BlackBerry OS. Lo que lo hace diferente es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma. El sistema permite programar aplicaciones en una variación de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, entre otras.) de una forma muy sencilla en un lenguaje de programación muy conocido como es Java (13).
- **iOS:** es conocido actualmente como el sistema operativo diseñado por Apple en California destinado para sus dispositivos de última generación (iPod, iPhone y iPad). El desarrollo originalmente se basa del núcleo sistema operativo Mac OS X, (DarwinBSD), pero a diferencia del sistema original, el ambiente gráfico del mismo es rediseñado para adaptarse a la plataforma móvil, permitiendo a su vez una mayor comodidad de uso y un sistema mucho más estable, a este nuevo

⁴Plugins: es una aplicación que se relaciona con otra para aportarle una función nueva.

entorno gráfico se le conoce como “SpringBoard” y este constantemente ha sido mejorado desde sus primeras versiones, las cuales aparecen con la integración del iPhone (14).

- **BlackBerry OS:** Blackberry OS (Operating System) es el sistema operativo multitarea desarrollado por la empresa Research In Motion Limited (RIM), una compañía de origen canadiense. Este software fue diseñado para desarrollarlo en una gama de dispositivos móviles denominados BlackBerry, surgidos a finales de los noventa. El gran distintivo del sistema operativo de BlackBerry es su diseño para un uso estrictamente profesional, basado en el administrador de contactos y con un gestor de correo electrónico. Destaca además, por disponer de un soporte para Java de MIDP 1.0 y MIDP 2.0; haciendo al dispositivo compatible para su sincronización con otros servidores de correo electrónico, tareas, calendario, etc (15).

Se decidió desarrollar la aplicación cliente del sistema SIIPOL Móvil para la plataforma Android, por las siguientes razones:

- Es uno de los sistemas operativos para móviles más usados en la actualidad (33).
- No es necesario pagar licencias a ninguna compañía para desarrollar y distribuir aplicaciones en Android.
- La programación en Android se realiza en lenguaje Java, uno de los lenguajes de uso más extendido y en el cual los autores de este trabajo tienen gran experiencia.
- Aunque Google es el mayor promotor de Android, existe un gran número de empresas que dan soporte a Android (16).

1.8 Plataforma de desarrollo

Java Enterprise Edition (JEE): el objetivo de la plataforma JEE es proporcionar a los desarrolladores con un potente conjunto de APIs, reducir el tiempo de desarrollo y la complejidad y mejorar el rendimiento de la aplicación.

La plataforma JEE utiliza un modelo de programación simplificado. Los descriptores de despliegue XML son opcionales. En su lugar, un desarrollador puede simplemente introducir la información como una anotación directamente en un archivo fuente de Java y el servidor JEE configurará el componente de despliegue y el tiempo de ejecución. Con anotaciones, la especificación de la información se coloca directamente en el código del programa que lo afecta. En la plataforma JEE, la inyección de dependencia se puede aplicar a todos los recursos que un componente necesita, ocultando la creación y búsqueda de recursos. La inyección de dependencia permite que el contenedor JEE inserte automáticamente referencias a otros componentes o recursos requeridos utilizando anotaciones (17).

1.9 Framework

Conjunto de clases que cooperan y forman un diseño reutilizable para un tipo específico de software. Un framework ofrece una guía arquitectónica partiendo del diseño en clases abstractas y definiendo sus responsabilidades y sus colaboraciones (18).

1.9.1 Hibernate versión 3

Hibernate facilita el almacenamiento y la recuperación de objetos de dominio de Java a través de Mapeo objeto / relacional. No sólo se ocupa de la asignación de clases Java a tablas de bases de datos (y de tipos de datos Java con tipos de datos SQL), sino que también proporciona consulta de datos y sistemas de recuperación.

Permite además desarrollar un modelo de datos utilizando todo el poder de un lenguaje orientado a objetos, incluyendo asociación, herencia, polimorfismo, composición. Está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a una base de datos ya existente. Hibernate simplifica el proceso de creación de aplicaciones y reduce drásticamente el tiempo de desarrollo, mantenimiento y resolución de errores (19).

1.9.2 Spring versión 3.1.2

Spring es un framework para construir aplicaciones empresariales. Spring está diseñado para no ser intrusivo, esto significa que no es necesario que la aplicación extienda o implemente alguna clase o interface de Spring, por lo que el código de lógica quedará libre y completamente reutilizable para un proyecto sin Spring, o por si se debe quitar de una aplicación que ya lo esté usando.

El núcleo de Spring está basado en un principio o patrón de diseño llamado Inversión de Control (IoC por sus siglas en inglés). Las aplicaciones que usan el principio de IoC se basan en su configuración que puede ser en archivos XML o con anotaciones para describir las dependencias entre sus componentes y los otros objetos con los que interactúa (20).

Entre sus características claves se encuentran:

- ✓ Implementa la filosofía POJO (Objeto Java), la cual revalora la simplicidad de las clases Java aportando manejo de transacciones de forma no intrusiva.
- ✓ Incorpora un módulo para la implementación de la seguridad, llamado Spring Security.
- ✓ Brinda soporte para el acceso y la publicación de servicios web.

1.10 Apache Tomcat versión 7.0.23

Apache Tomcat es un contenedor web de código abierto escrito en Java, por lo que funciona en cualquier sistema operativo que disponga de una máquina virtual Java. Fue desarrollado en un ambiente

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

participativo y abierto. Incluye muchas características que lo convierten en una plataforma útil para el desarrollo y despliegue de aplicaciones web y servicios web (32):

- Soporte para la inclusión de contenidos externos directamente en una aplicación web.
- Mejoras para detectar y prevenir "fugas de memoria" en las aplicaciones web.
- Limpieza interna de código.

Conclusiones Parciales

En este capítulo se presentaron y analizaron varios software de gestión de la información policial, llegándose a la conclusión que con ninguno de ellos podía resolverse la situación problemática del CICPC, siendo necesaria la implementación de un nuevo sistema. Se seleccionaron las herramientas de desarrollo para la implementación de la aplicación. Se seleccionó Android como plataforma de desarrollo para el cliente móvil y se presentaron varias razones para esta selección. Se seleccionó también la metodología de desarrollo y se expusieron sus principales características.

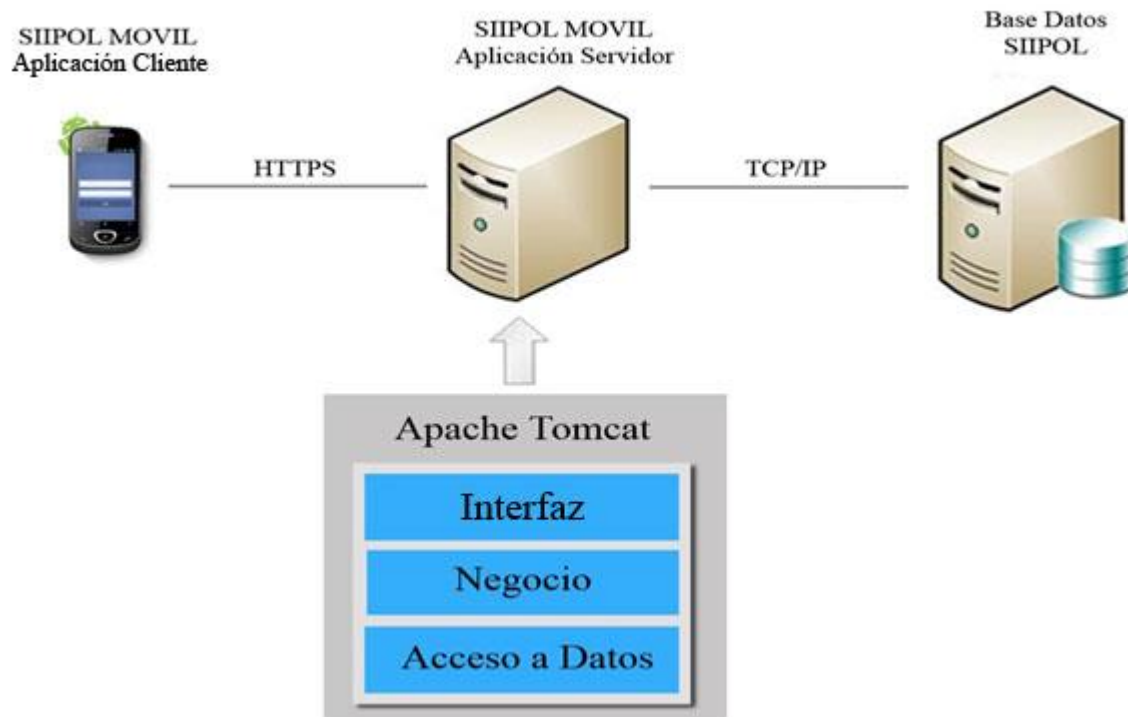
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.1 Introducción

En el presente capítulo se realiza el Análisis y Diseño a la propuesta del sistema. En el análisis se especifican los requisitos funcionales y no funcionales que el sistema debe cumplir para satisfacer las necesidades del cliente, además, se identifican las personas relacionadas con el sistema.

2.2 Propuesta de Solución

La propuesta de solución se basa en una arquitectura cliente-servidor, el cliente es una aplicación móvil para el sistema operativo Android llamada SIIPOL MOVIL Aplicación Cliente. La aplicación servidor llamada, SIIPOL MOVIL Aplicación Servidor, utiliza el servidor web Apache Tomcat. La arquitectura de la aplicación servidor se basa en el patrón N Capas, cuenta con una capa de Servicios, una capa de Negocio y una capa Acceso a Datos. El intercambio de datos con el cliente se realizara a través de servicios REST y para garantizar la seguridad de los datos se empleará el protocolo HTTPS. La conexión a la base de datos se realiza a través del protocolo TCP/IP. El sistema funcionará de acuerdo a la Figura 1.



2.3 Requerimientos del Sistema

Los requerimientos de un software son las propiedades o restricciones, determinadas con precisión, que un producto software debe poseer. Los mismos se clasifican en funcionales y no funcionales. Los requisitos funcionales son las condiciones que debe cumplir el sistema, mientras que los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades son las que hacen al producto atractivo, usable, rápido o confiable (10).

2.3.1 Requisitos no funcionales

Requisitos no funcionales de la aplicación Cliente

Usabilidad:

RnF1. El sistema podrá ser utilizado por cualquier usuario con las siguientes características:

- ✓ Conocimientos básicos relativos al uso de un dispositivo móvil.
- ✓ Conocimientos sólidos relativos a los procesos de negocio acorde al rol que desempeñe.

RnF2. Debido a que la aplicación será desplegada en un país hispanohablante, todas las interfaces del sistema así como los mensajes de errores y notificaciones deberán ser en idioma español.

Fiabilidad:

RnF3. El sistema permitirá realizar consultas en todo momento, siempre que se pueda establecer comunicación con el servidor.

RnF4. La aplicación cliente nunca accederá directamente a la base de datos, sino a través de la aplicación servidor.

Rendimiento:

RnF5. La respuesta del sistema estará en correspondencia con las tecnologías de acceso al servicio, lo cual depende de la red telefónica y la carga del servidor.

Seguridad:

RnF6. Confidencialidad: Los usuarios se autentican para poder acceder al sistema según los permisos asignados.

Requisitos de hardware:

RnF7. Los teléfonos móviles donde se ejecuta la aplicación cliente deben tener los elementos de hardware necesarios para poder establecer una conexión con el servidor donde este desplegada la aplicación servidor.

CAPÍTULO 2: propuesta de solución

Requisitos de software:

RnF8. El sistema operativo necesario en el teléfono móvil debe ser Android en su versión v2.2 como mínimo.

Requisitos no funcionales de la aplicación Servidor

Fiabilidad

RnF 1. Los clientes accederán a los datos de la base de datos a través de la aplicación, nunca harán consultas de forma directa a la BD.

Usabilidad

RnF 2. Los resultados de las consultas de las entidades operativas que tengan más de 5 coincidencias serán paginados.

Seguridad

RnF 3. El sistema concederá acceso a partir de un nombre de usuario y una contraseña.

RnF 4. El sistema concederá el acceso a cada usuario autenticado solamente a las funciones que le estén permitidas, de acuerdo a su perfil.

RnF 5. Las conexiones serán a través del protocolo HTTPS.

RnF 6. El sistema inhabilitará los accesos de los usuarios que realice 5 o más intentos fallidos de acceso al sistema.

RnF 7. El sistema implementará el uso de validaciones para garantizar la integridad de la información que se introduce por el usuario.

RnF 8. Todo el intercambio de datos entre la aplicación servidor y la base de datos del CICPC se realizara a través de transacciones, de esta forma, se evita la existencia de datos corruptos o incompletos.

RnF 9. El sistema solo tendrá acceso de escritura en un esquema propio. Será imposible para el sistema modificar datos en los esquemas que contienen información sensitiva

Rendimiento

RnF 10. Las operaciones que impliquen un elevado nivel procesamiento se implementarán usando procedimientos almacenados.

Restricciones de diseño

RnF 11. El sistema se implementará usando la plataforma JEE.

RnF 12. El sistema estará basado en un estilo arquitectónico en capas.

CAPÍTULO 2: propuesta de solución

RnF 13. La comunicación entre los clientes y el sistema se realizará a través de servicios web tipo REST.

Personas relacionadas con el Sistema

La información sólo puede ser accedida y modificada por las personas autorizadas y de la forma autorizada, lo que significa que se definirán roles para acceder al sistema, estos roles deben asignarse adecuadamente a los diferentes tipos de personas, según sus capacidades y puestos de cada una de ellas. La interacción directa con el sistema lo realizarán los siguientes tipos de persona:

Persona encargada	Descripción
Funcionario	Agente autorizado a consultar información operativa.
Administrador	Agente autorizado a gestionar roles y privilegios, también autorizado a consultar información operativa.

Tabla 1. Descripción de las personas relacionadas con el Sistema

2.4 Exploración

El ciclo de vida de un proyecto realizado con la metodología XP se inicia con la Exploración. El cliente plantea a grandes rasgos las historias de usuarios. Al finalizar el equipo cuenta con suficiente material de trabajo como para producir una primera entrega. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas y tecnologías que serán utilizadas en el proyecto (10).

Historias de usuario (HU)

Las historias de usuario corresponden a la técnica utilizada para especificar los requisitos del software. Se trata de formatos en los cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (10).

A continuación se muestra la Historia de Usuario correspondiente a Registrar Traza de Consulta Sobre Entidad Operativa. Para ver las restantes dirigirse al **Anexo 1**.

Historia de Usuario

CAPÍTULO 2: propuesta de solución

Número: 13	Usuario:
Nombre historia: Registrar Traza de Consulta Sobre Entidad Operativa.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Carlos León Castellanos	
Descripción: Se registra una traza con los datos de una consulta sobre una entidad operativa.	
Observaciones: Debe haberse realizado una consulta sobre una entidad operativa.	

Tabla 2.Registrar Traza de Consulta Sobre Entidad Operativa.

2.5Planificación.

El cliente establece la prioridad que tendrá cada historia de usuario según sus necesidades más inmediatas, luego los programadores realizan una estimación del esfuerzo que se necesita para cada una de ellas. Además se toman acuerdos sobre el contenido de las entregas (10).

Estimación de esfuerzo por HU.

Las estimaciones de esfuerzo asociado a la implementación de las historias de usuario las establecen los programadores, para ello utilizan puntos. Un punto, equivale a una semana ideal de programación (10). Para el logro del sistema se ha realizado la estimación de esfuerzo por cada historia de usuario, las que se muestran a continuación:

No.	Historia de Usuario	Puntos de estimación
1.	Consultar Personas con Registros Policiales.	1
2.	Ver Detalles de Personas con Registros Policiales.	0.5
3.	Consultar Vehículos con Registros Policiales.	1
4.	Ver Detalles de Vehículos con Registros Policiales.	0.5
5.	Consultar Vehículos INTT	1
6.	Ver Detalles de Vehículos INTT	0.5
7.	Consultar Armas con Registros Policiales.	1

CAPÍTULO 2: propuesta de solución

8.	Ver Detalles de Armas con Registros Policiales.	0.5
9.	Consultar Objetos con Registros Policiales.	1
10.	Ver Detalles de Objetos con Registros Policiales.	0.5
11.	Consultar Persona Natural.	1
12.	Ver Detalles de Persona Natural.	0.5
13.	Registrar Traza de Consulta Sobre Entidad Operativa.	1
14.	Crear Rol	0.5
15.	Modificar Rol	0.5
16.	Eliminar Rol.	0.5
17.	Crear Usuario	0.5
18.	Modificar Usuario	0.5
19.	Buscar Usuario	0.5
20.	Eliminar Usuario	0.5
21.	Autenticar	0.5

Tabla 3. Estimación de esfuerzo por HU

2.6 Plan de iteraciones.

El plan de iteraciones muestra cuales son las historias de usuarios que serán implementadas en cada iteración del sistema y las posibles fechas de liberaciones.

En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el transcurso del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente es quien decide cuales son las historias que se implementarán en cada iteración. Al final de la última iteración el sistema estará listo para entrar en producción. Tratando de esta manera de tener preparadas las funcionalidades básicas e indispensables del sistema (10).

Se decide realizar el sistema en tres iteraciones las cuales se detallan a continuación:

Iteración 1: En la iteración 1 serán implementadas las HU que permiten la consulta de la entidades operativa necesarias así como las trazas que se guardan una vez que realiza una consulta, estas HU son la 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 y 13; las mismas tiene una prioridad alta en el negocio por su relevancia, obteniéndose así la versión 0.1 del producto, dándole la posibilidad al cliente de probar dichas funcionalidades, que son de carácter crítico para él.

CAPÍTULO 2: propuesta de solución

Iteración 2: En esta iteración se implementan las historias de usuario con prioridad media, estas nos brindan la posibilidad de administrar los usuarios, roles y permisos, así como permitir la autenticación y tener el control de quienes acceden a la aplicación. Al finalizar, se tendrán implementadas las historias de usuario: 14, 15, 16, 17, 18, 19, 20 y 21. Como resultado de esta iteración se contará con la versión 0.2 del producto final.

2.7 Plan de duración de iteraciones.

Como parte del ciclo de vida de un proyecto utilizando la metodología XP se crea el plan de duración de iteraciones. Para la confección del plan se tiene en cuenta la estimación de esfuerzo para cada HU. Este plan permite mostrar la duración de cada iteración así como el orden en que se implementarán las historias de usuario, teniendo una mayor organización (10).

Iteración	Historia de Usuario	Duración (Semanas)
Iteración 1	Consultar Personas con Registros Policiales.	10 semanas
	Ver Detalles de Personas con Registros Policiales.	
	Consultar Vehículos con Registros Policiales.	
	Ver Detalles de Vehículos con Registros Policiales.	
	Consultar Vehículos INTT	
	Ver Detalles de Vehículos INTT	
	Consultar Armas con Registros Policiales.	
	Ver Detalles de Armas con Registros Policiales.	
	Consultar Objetos con Registros Policiales.	
	Ver Detalles de Objetos con Registros Policiales.	
	Consultar Persona Natural.	
	Ver Detalles de Persona Natural.	
	Registrar Traza de Consulta Sobre Entidad Operativa.	
Iteración 2	Crear Rol	4 semanas
	Modificar Rol	
	Eliminar Rol.	
	Crear Usuario	
	Modificar Usuario	

CAPÍTULO 2: propuesta de solución

	Buscar Usuario	
	Eliminar Usuario	
	Autenticar	

Tabla 4. Plan de Duración de Iteraciones

2.8 Plan de entregas

Con el plan de entrega se da un aproximado de las entregas de las versiones contando con que la implementación comienza en el mes de marzo:

A entregar	Final 1ra iteración	Final 2da iteración
Aplicación Cliente SIIPOL MÓVIL	15/4/2013 Versión 0.1	16/5/2013 Versión 0.2

Tabla 5. Plan de entregas

Conclusiones Parciales

Durante el desarrollo de este capítulo se desarrolló la propuesta de solución que se desea implementar. Para esto se definieron los requisitos no funcionales y las HU propuestas por el cliente. Se realizó la estimación de esfuerzo necesaria para cada HU, y por último se construyeron el Plan de duración de las iteraciones y el Plan de entregas.

CAPÍTULO 3: DISEÑO

3.1 Introducción

Luego de las fases de Exploración y Planificación explicadas en el capítulo anterior damos paso a la fase de diseño del sistema. Se define el patrón de arquitectura a seguir, así como los patrones de diseño a utilizar para la implementación del sistema.

3.2 Arquitectura de la solución

La solución está basada en una arquitectura Cliente/Servidor, que separa los módulos del *software*, brindando una mayor usabilidad, flexibilidad, interoperabilidad y escalabilidad en las comunicaciones. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita el diseño del sistema. Al remitente de una solicitud es conocido como Cliente y al receptor de la solicitud enviada por el cliente se conoce como Servidor (21).

3.3 Patrón arquitectónico N Capas

Se propone la utilización del patrón arquitectónico N Capas que tiene como objetivo principal separar los diferentes aspectos del desarrollo, tales como las cuestiones de lógica de negocio y los mecanismos de almacenamiento. Los componentes de cada capa se comunican con los componentes de otras capas a través de interfaces bien conocidas, cada nivel agrega las responsabilidades y abstracciones del nivel inferior (22).

3.3.1 Principios claves

- ✓ Separa de forma clara la funcionalidad de cada capa.
- ✓ Cada capa contiene la funcionalidad relacionada solo con las tareas de esa capa.
- ✓ Las capas inferiores no tienen dependencias de las capas superiores.
- ✓ La comunicación entre capas está basada en una abstracción que proporciona bajo acoplamiento entre capas (22).

3.3.2 Beneficios

- ✓ Aislamiento: se pueden realizar actualizaciones en el interior de las capas sin que esto afecte el resto del sistema.
- ✓ Abstracción: los cambios se realiza a alto nivel y se puede incrementar o reducir el nivel de abstracción que se usa en cada capa del modelo.

- ✓ Facilidad para las pruebas: Ya que cada capa tiene una interfaz bien definida sobre la que realizar las pruebas y la habilidad de cambiar entre diferentes implementaciones de una capa (22).

3.4 Patrones de diseño.

Un patrón de diseño es una buena práctica documentada, o solución, que se ha aplicado con éxito en ambientes múltiples para solucionar un problema que se repite en los sistemas de dichos ambientes y en situaciones específicas (23).

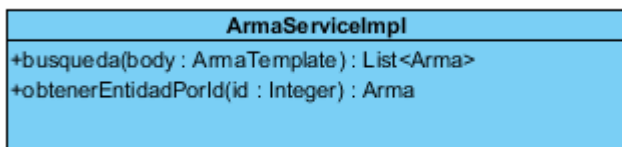
3.4.1 Patrones para asignar responsabilidad.

Los patrones describen los elementos esenciales de la asignación de responsabilidades a objetos, donde son expresados en forma de patrones (24).

Se emplearon en el sistema los siguientes patrones GRASP:

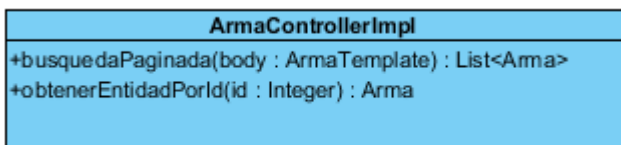
➤ **Experto**

Este patrón se encarga de asignar una responsabilidad al experto en información, o sea, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad. En la solución las clases del negocio realizan las validaciones correspondientes a las peticiones de los usuarios y las de acceso a datos se encargan de la gestión de los datos entregados desde la lógica del negocio.



➤ **Controlador**

Este patrón asigna la responsabilidad del manejo de los eventos de un sistema a una o varias clases. Cada petición que se genera es manejada por una clase controladora, ésta invoca a las clases de su negocio que necesita para cumplir con la petición, recibe la respuesta y ofrece una respuesta para mostrar finalmente la respuesta al usuario.



➤ **Alta cohesión y Bajo acoplamiento**

La cohesión es la medida de la fuerza que une a las responsabilidades de una clase. Una clase con alta cohesión mejora la claridad y la facilidad de su uso, su mantenimiento se simplifica y es fácil de reutilizar. A menudo, se genera un bajo acoplamiento. Este consiste en tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización y disminuyendo la dependencia entre las clases. El sistema fue diseñado de manera tal que las clases pertenecientes a las diferentes capas se le asignaran las responsabilidades necesarias y bien delimitadas para garantizar una alta cohesión.

3.4.2 Patrones GoF

Los patrones GoF son una forma de enfrentarse a los problemas típicos de la programación (24).

➤ **Singleton**

Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. El marco de trabajo Spring lo utiliza a través de la inyección de dependencia que permite tener acceso a dicha instancia desde cualquier clase.

```
@Autowired
private PerfilDao perfilDao;
```

➤ **Observador**

El patrón Observador (Observer) o también conocido como el patrón publicación-subscripción, define una relación de uno a muchos objetos, de manera que cuando uno de los objetos cambia su estado, el observador se encarga de notificar este cambio a todos los otros objetos.

3.5 Tarjetas Clase-Responsabilidad-Colaboración (CRC)

La metodología XP no obliga a la realización de diagramas UML para representar las clases que contendrán la lógica del negocio, en su lugar propone el uso de tarjetas CRC como técnica de modelado para ayudar a los desarrolladores de software a crear diseños de clases orientados a responsabilidades. Las tarjetas CRC permiten trabajar con una metodología orientada a objetos, cada tarjeta representa un objeto (10).

El título de la tarjeta corresponde al nombre de la clase, las Responsabilidades son colocadas en la izquierda y las Colaboraciones en la parte derecha como se muestra a continuación:

3.5.1 Tarjetas CRC para el cliente

Clase: ConsultarArma	
Descripción: Clase encargada de la consulta sobre las armas.	
Responsabilidad	Colaborador
Consultar armas	ParserAsyncTask ResultadosArma

Tabla 6. Tarjetas CRC para el cliente

3.5.2 Tarjetas CRC para el servidor

Clase: TrazaDaolmpl	
Descripción: Clase encargada de salvar trazas en la base de datos.	
Responsabilidad	Colaborador
Salvar las trazas de consulta de entidades operativas	
Salvar las trazas de gestión de usuarios y perfiles	

Tabla 7. Tarjetas CRC para el servidor

Las demás Tarjetas CRC se encuentran en el **Anexo 3**.

Conclusiones Parciales

En este capítulo se realizó el análisis del sistema, se definieron los patrones arquitectónicos y de diseño usados con el objetivo de lograr una mayor organización en los elementos que conforman la aplicación así como también se confeccionaron las Tarjetas CRC que son fundamentales para la implementación del sistema.

CAPÍTULO 4: Implementación y prueba

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción

En el presente capítulo se documentan las fases de implementación y prueba según lo propone la metodología de desarrollo XP. Se desglosan las HU en tareas con el objetivo de facilitar el trabajo de los desarrolladores. Se muestran los resultados de las pruebas de aceptación realizadas a la aplicación.

4.2 Tareas de Ingeniería

Las Tareas de Ingeniería (TI) se realizan para llevar a cabo la correcta implementación de las HU descritas por el cliente, que se realizan en cada una de las iteraciones. Las TI permiten a los desarrolladores obtener un nivel de detalle más avanzado que el que propicia las HU.

El desarrollo del software se planificó en dos iteraciones de trabajo. Para la implementación de las HU de estas iteraciones se definieron las TI como se muestra a continuación:

Tarea de Ingeniería	
Número: 1	Historia de usuario No: 1
Nombre de Tarea: Consultar Personas con Registros Policiales.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Descripción: Permite al usuario consultar información operativa de personas con registros policiales dependiendo de los criterios de búsqueda especificados y se devuelve una lista de coincidencias de personas con registros policiales.	

Tabla 6. Tarea de Ingeniería: Consultar Personas con Registros Policiales

Para ver las demás TI, ver **Anexo 2**.

4.3 Validación de la Propuesta

Una vez concluida la implementación del sistema se desarrollaron una serie de pruebas para comprobar que se cumplieron los requerimientos descritos por el cliente.

La metodología XP propone las pruebas de unidad como parte constante del proceso de desarrollo (25). Para la implementación del sistema no se hicieron pruebas de unidad debido a problemas de planificación que conllevaron a una disminución en el tiempo disponible para el desarrollo. Después de una negociación

con el cliente, se acordó sustituir las pruebas de unidad por pruebas de aceptación al final de cada iteración.

4.3.1 Las pruebas de aceptación

El objetivo de estas pruebas es verificar los requisitos, por este motivo, los propios requisitos del sistema son la principal fuente de información a la hora de construir las pruebas de aceptación.

Las pruebas de aceptación son creadas a partir de las historias de usuario. Durante una iteración la historia de usuario seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación. El cliente o usuario especifica los aspectos a testear cuando una historia de usuario ha sido correctamente implementada (25).

A continuación se muestran los resultados de las pruebas de aceptación realizadas a las HU pertenecientes a la primera iteración del software.

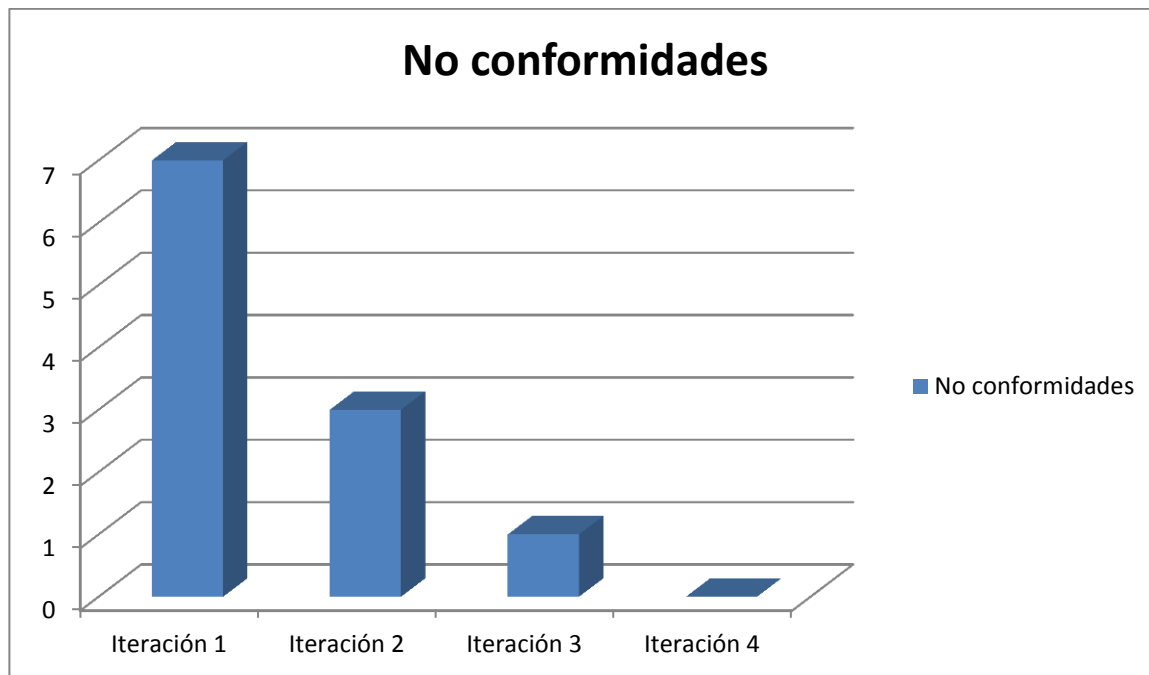


Tabla 7: Resultados de las Pruebas de Aceptación realizadas en la primera iteración

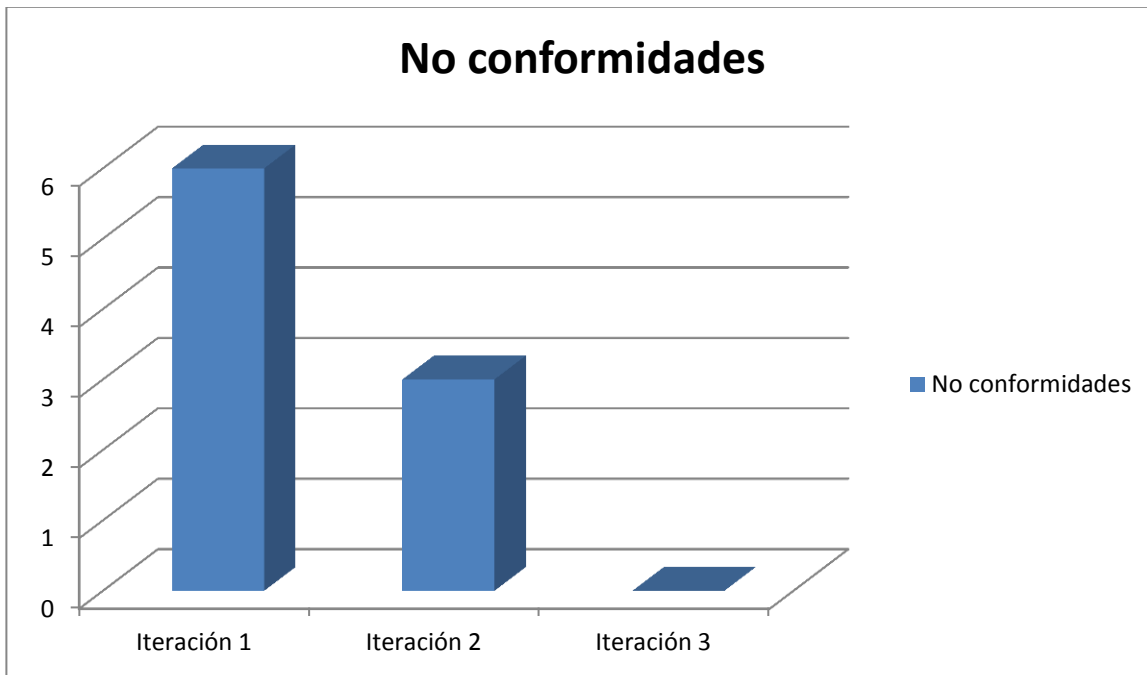


Tabla 8: Resultados de las Pruebas de Aceptación realizadas en la segunda Iteración

Conclusiones Parciales

En este capítulo se generaron los artefactos propuestos por la metodología XP para las fases de Implementación y Prueba. Se realizaron las tareas de ingeniería que dieron solución a todas las HU logrando una mayor organización y rapidez en el desarrollo del sistema. Se mostraron los resultados de las pruebas de aceptación realizadas para comprobar que se cumplieron todos los objetivos trazados.

CONCLUSIONES GENERALES

Con el propósito de darle cumplimiento al objetivo general y a la problemática planteada en el presente trabajo, se han llevado a cabo satisfactoriamente cada una de las tareas que fueron trazadas al comienzo del mismo.

Durante el desarrollo de la investigación, se profundizó en el conocimiento de las herramientas utilizadas, así como en la metodología para guiar el proceso de desarrollo del software, con la cual se generaron los artefactos necesarios para validar la solución. Se puede concluir por tanto, que se ha cumplido satisfactoriamente el objetivo trazado para este trabajo, enfatizando en los siguientes puntos:

- Se eligieron las tecnologías adecuadas para desarrollar el sistema, cumpliendo con los requisitos planteados por el cliente.
- Se desarrolló una capa de servicios REST para la búsqueda de información operativa en las bases de datos del CICPC.
- Se desarrolló un subsistema para la administración de usuarios y permisos de la aplicación.
- Se implementó un cliente para móviles Android, que permite consultar la información operativa del CICPC.

Teniendo en cuenta los resultados obtenidos, se concluye que los objetivos presentados al inicio de la investigación han sido cumplidos en su totalidad.

RECOMENDACIONES

Una vez cumplidos los objetivos de este trabajo, y teniendo en cuenta las experiencias obtenidas durante la realización del mismo, se recomienda:

- Desarrollar el cliente para otros dispositivos móviles, como los que utilizan BlackBerry OS y iOS.
- Desarrollar un sistema similar para los órganos del MININT en Cuba.

Referencias Bibliográficas

1. **Misionescuatro.com.** Misionescuatro.com. [En línea] 13 de 6 de 2013. [Citado el: 13 de 6 de 2013.] <http://www.misionescuatro.com/ampliar.php?id=27960>.
2. **Consortium, World Wide Web.** World Wide Web Consortium (W3C) - España. [En línea] [Citado el: 18 de 1 de 2013.] <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
3. **Fielding, Roy Thomas.** *Architectural Styles and the Design of Network-based Software Architectures*. 2000.
4. **Amodeo, Enrique.** ¿Qué es REST? [En línea] 7 de 2010. <http://eamodeorubio.wordpress.com/2010/07/26/servicios-web-2-%C2%BFque-es-rest/>.
5. **Rodriguez, Alex.** RESTful Web services: The basics. [En línea] 11 de 2008. <http://www.ibm.com/developerworks/webservices/library/ws-restful/index.html>.
6. **Mobile, WM Wireles &.** WM Wireles & Mobile. [En línea] [Citado el: 19 de 1 de 2013.] <http://www.wi-mobile.com/es/soluciones/gobierno/policia/policia/wm-police-solucion-para-pago-movil-de-multas-y-comparendos>.
7. **TECHNOLOGIES, CROSS MATCH.** CROSS MATCH TECHNOLOGIES. [En línea] [Citado el: 19 de 1 de 2013.] <http://www.crossmatch.com/spanish/news/Mobile-Rapid-ID-Certified-by-FDLE.php>.
8. **COMPUTERWORLD.** COMPUTERWORLD. [En línea] [Citado el: 19 de 1 de 2013.] <http://www.caracol.com.co/noticias/tecnologia/solucion-blackberry-ayuda-a-luchar-contra-el-crimen-y-protoger-la-comunidad-en-brasil/20120503/nota/1681667.aspx>.
9. **kent, Beck.** *Extreme Programming Explained*. Boston : s.n., 2000.
10. **Gimson, Loraine.** *Metodologías ágiles y desarrollo basado en conocimientos*. 2012.
11. **Alvarez, Miguel Angel.** desarrolloweb.com. [En línea] [Citado el: 21 de 1 de 2013.] <http://www.desarrolloweb.com/articulos/497.php>.
12. **Belmonte Fernández, Oscar, Granell Canut, carlos y Erdozain Navarro, maria Del Carmen.** *Desarrollo de Proyectos Informáticos con Tecnología Java*.
13. **xatakandroid.** xatakandroid. [En línea] 8 de 2 de 2011. [Citado el: 22 de 1 de 2013.] <http://www.xatakandroid.com/sistema-operativo/que-es-android>.
14. **Asorce.** Asorce. [En línea] 29 de 4 de 2012. [Citado el: 23 de 1 de 2013.] <http://www.osupiita.com/index.php/proyectos/ios/que-es-ios>.
15. **MANUALES.com.** MANUALES.com. [En línea] [Citado el: 23 de 1 de 2013.] <http://www.manuales.com/manual-de/que-es-blackberry-os>.
16. **Manuel, Báez, y otros.** *Introducción a Android*. s.l. : E.M.E. Editorial ©.
17. **Jendrock, Eric, y otros.** *The JavaEE6 Tutorial, Volume1 Basic Concepts*. 2009.
18. **Framework.** Framework. [En línea] [Citado el: 24 de 1 de 2013.] <http://www.framework.com/>.
19. **Rosés Albiol, Francesc.** *Introducción a Hibernate*. 2003.
20. **Johnson, Rod, y otros.** *Spring Reference Documentation 3.1*. 2012.
21. **MASTERMAGAZINE.** MASTERMAGAZINE. [En línea] [Citado el: 5 de 2 de 2013.] <http://www.mastermagazine.info/termino/4294.php>.
22. **de la Torre , César Llorente, y otros.** *Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0*. 2010.
23. **S. Pressman, Roger.** *Ingeniería de Software un enfoque práctico*. 2005.

24. **Larman, Craig** . *UML y Patrones*. 2004.
25. **Rodríguez Corbea, Maite y Ordóñez Pérez, Meylin** . *LA METODOLOGÍA XP APLICABLE AL DESARROLLO DEL SOFTWARE EDUCATIVO EN CUBA*. Ciudad de la Habana : s.n., 2007.
26. **Ingeniatic**. Ingeniatic. [En línea] 2011. [Citado el: 14 de 1 de 2013.]
<http://ingeniatic.euitt.upm.es/index.php/multimedia/infografias/item/701-c%C3%B3mo-funcionan-las-comunicaciones-con-tel%C3%A9fonos-m%C3%B3viles>.
27. **GSMspain**. GSMspain. [En línea] [Citado el: 15 de 1 de 2013.]
<http://www.gsmspain.com/glosario/?palabra=GPRS>.
28. **Movistar**. Escritorio Movistar Latam. [En línea] [Citado el: 15 de 1 de 2013.]
http://www.movistar.cl/PortalMovistarWeb/ShowDoc/WLP+Repository/Portlets/P030_Generico/Documents/Manual_Linux..
29. **Angulo Martínez, Jorge Luis y Gómez García, Arsenio David** . *DESARROLLO DE UNA HERRAMIENTA DE MEDICIÓN DE DESEMPEÑO EN TRANSMISIÓN DE DATOS PARA UNA RED CDMA 1xEVDO*. Caracas : s.n., 2006.
30. **ENTER.CO**. ENTER.CO. [En línea] 2 de 11 de 2012. [Citado el: 15 de 1 de 2013.]
<http://www.enter.co/colombiadigital/las-13-cosas-que-usted-tiene-que-saber-sobre-4g/>.
31. **CIENTEC**. CIENTEC. [En línea] [Citado el: 15 de 1 de 2013.]
<http://www.cientec.com/tendencias/tendencias-wimax.html>.
32. Apache Tomcat. *Apache Tomcat*. [En línea] [Citado el: 2013 de 6 de 7.] <http://tomcat.apache.org/>.
33. **Canalys**. Canalys. [En línea] 9 de 5 de 2013. [Citado el: 12 de 6 de 2013.]
<http://www.canalys.com/newsroom/smart-mobile-device-shipments-exceed-300-million-q1-2013>.

ANEXOS

ANEXO 1: Historias de Usuarios

Historia de Usuario	
Número: 1	Usuario: Funcionario, Administrador
Nombre historia: Consultar Personas con Registros Policiales.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Carlos león Castellanos	
Descripción: Permite al usuario consultar información operativa de personas con registros policiales dependiendo de los criterios de búsqueda especificados y se devuelve una lista de coincidencias de personas con registros policiales.	
Observaciones: Debe haberse autenticado y el usuario debe tener permiso para realizar consultas sobre personas con registros policiales.	

Tabla 5. HU 1 - Consultar Personas con Registros Policiales.

Historia de Usuario	
Número: 2	Usuario: Funcionario, Administrador
Nombre historia: Ver Detalles de Personas con Registros Policiales	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Carlos león Castellanos	
Descripción: Devuelve solo los datos que aparezcan en el sistema de la persona seleccionada.	
Observaciones:	

Tabla 6. HU 2 - Ver Detalles de Personas con Registros Policiales

Historia de Usuario	
Número: 3	Usuario: Funcionario, Administrador
Nombre historia: Consultar Vehículos con Registros Policiales.	

Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Carlos león Castellanos	
Descripción: Permite al usuario consultar información operativa de vehículos con registros policiales dependiendo de los criterios de búsqueda especificados y se devuelve una lista de coincidencias de vehículos con registros policiales.	
Observaciones: Debe haberse autenticado y el usuario debe tener permiso para realizar consultas sobre vehículos con registros policiales.	

Tabla 7. HU 3 - Consultar Vehículos con Registros Policiales.

Historia de Usuario	
Número: 4	Usuario: Funcionario, Administrador
Nombre historia: Ver Detalles de Vehículos con Registros Policiales.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Carlos león Castellanos	
Descripción: Devuelve solo los datos que aparezcan en el sistema del vehículo seleccionado.	
Observaciones:	

Tabla 8. HU 4 - Ver Detalles de Vehículos con Registros Policiales.

Historia de Usuario	
Número: 5	Usuario: Funcionario, Administrador
Nombre historia: Consultar Vehículos INTT.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Carlos león Castellanos	
Descripción: Permite al usuario consultar información operativa de vehículos INTT	

dependiendo de los criterios de búsqueda especificados y se devuelve una lista de coincidencias de vehículos con registros policiales.
Observaciones: Debe haberse autenticado y el usuario debe tener permiso para realizar consultas sobre vehículos INTT.

Tabla 9. HU 5 - Consultar Vehículos INTT.

Historia de Usuario	
Número: 6	Usuario: Funcionario, Administrador
Nombre historia: Ver Detalles de Vehículos INTT.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Carlos león Castellanos	
Descripción: Devuelve solo los datos que aparezcan en el sistema del vehículo seleccionado.	
Observaciones:	

Tabla 10. HU 6 - Ver Detalles de Vehículos INTT.

Historia de Usuario	
Número: 7	Usuario: Funcionario, Administrador
Nombre historia: Consultar Armas con Registros Policiales.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Carlos león Castellanos	
Descripción: Permite al usuario consultar información operativa de armas con registros policiales dependiendo de los criterios de búsqueda especificados y se devuelve una lista de coincidencias de vehículos con registros policiales.	
Observaciones: Debe haberse autenticado y el usuario debe tener permiso para realizar consultas sobre armas con registros policiales.	

Tabla 11. HU 7 - Consultar Armas con Registros Policiales.

Historia de Usuario	
Número: 8	Usuario: Funcionario, Administrador
Nombre historia: Ver Detalles de Armas con Registros Policiales.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Carlos león Castellanos	
Descripción: Devuelve solo los datos que aparezcan en el sistema del arma seleccionada.	
Observaciones:	

Tabla 12. HU 8 - Ver Detalles de Armas con Registros Policiales.

Historia de Usuario	
Número: 9	Usuario: Funcionario, Administrador
Nombre historia: Consultar Objetos con Registros Policiales.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Carlos león Castellanos	
Descripción: Permite al usuario consultar información operativa de objetos con registros policiales dependiendo de los criterios de búsqueda especificados y se devuelve una lista de coincidencias de vehículos con registros policiales.	
Observaciones: Debe haberse autenticado y el usuario debe tener permiso para realizar consultas sobre objetos con registros policiales.	

Tabla 13. HU 9 - Consultar Objetos con Registros Policiales.

Historia de Usuario	
Número: 10	Usuario: Funcionario, Administrador
Nombre historia: Ver Detalles de Objetos con Registros Policiales.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media

Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Carlos León Castellanos	
Descripción: Devuelve solo los datos que aparezcan en el sistema del objeto seleccionado.	
Observaciones:	

Tabla 14. HU 10 - Ver Detalles de Objetos con Registros Policiales.

Historia de Usuario	
Número: 11	Usuario: Funcionario, Administrador
Nombre historia: Consultar Persona Natural.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Reynier Moreu Morell	
Descripción: Permite al usuario consultar información operativa de personas naturales dependiendo de los criterios de búsqueda especificados y se devuelve una lista de coincidencias de personas naturales.	
Observaciones: Debe haberse autenticado y el usuario debe tener permiso para realizar consultas sobre personas naturales.	

Tabla 15. HU 11 - Consultar Persona Natural.

Historia de Usuario	
Número: 12	Usuario: Funcionario, Administrador
Nombre historia: Ver Detalles de Persona Natural.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Reynier Moreu Morell	
Descripción: Devuelve solo los datos que aparezcan en el sistema de la persona natural seleccionada.	

Observaciones:

Tabla 16. HU 12 - Ver Detalles de Persona Natural.

Historia de Usuario	
Número: 14	Usuario: Administrador
Nombre historia: Crear Rol.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Reynier Moreu Morell	
Descripción: Le permitirá al Administrador crear un rol.	
Observaciones: Debe haberse autenticado y el usuario debe tener permiso para crear rol.	

Tabla 17. HU 14 - Crear Rol.

Historia de Usuario	
Número: 15	Usuario: Administrador
Nombre historia: Modificar Rol.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Reynier Moreu Morell	
Descripción: Le permitirá al Administrador modificar un rol determinado.	
Observaciones: Debe haberse autenticado y el usuario debe tener permiso para modificar rol	

Tabla 18. HU 15 - Modificar Rol.

Historia de Usuario	
Número: 16	Usuario: Administrador
Nombre historia: Eliminar Rol.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media

Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Reynier Moreu Morell	
Descripción: Le permitirá al Administrador eliminar un rol.	
Observaciones: Debe haberse autenticado y el usuario debe tener permiso para eliminar rol.	

Tabla 19. HU 16 - Eliminar Rol.

Historia de Usuario	
Número: 17	Usuario: Administrador
Nombre historia: Crear Usuario	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Reynier Moreu Morell	
Descripción: Le permitirá al Administrador crear un usuario y asignarle los roles correspondiente.	
Observaciones:	

Tabla 20. HU 17 - Crear Usuario.

Historia de Usuario	
Número: 18	Usuario: Administrador
Nombre historia: Modificar Usuario	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Reynier Moreu Morell	
Descripción: Le permitirá al Administrador modificar un usuario existente en el sistema, así como sus roles.	
Observaciones:	

Tabla 21. HU 18 - Modificar Usuario.

Historia de Usuario

Número: 19	Usuario: Administrador
Nombre historia: Buscar Usuario	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Reynier Moreu Morell	
Descripción: Le permitirá al Administrador buscar un usuario existente en el sistema.	
Observaciones:	

Tabla 22. HU 19 - Buscar Usuario.

Historia de Usuario	
Número: 20	Usuario: Administrador
Nombre historia: Eliminar Usuario.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Reynier Moreu Morell	
Descripción: Le permitirá al Administrador eliminar un usuario existente en el sistema.	
Observaciones:	

Tabla 23. HU 20 - Eliminar Usuario.

Historia de Usuario	
Número: 21	Usuario: Funcionario, Administrador
Nombre historia: Autenticar	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Reynier Moreu Morell	
Descripción: Le permitirá al usuario autenticarse en la aplicación.	
Observaciones:	

Tabla 24. HU 21 – Autenticar.

Anexo 2: Tareas de ingenierías

Tarea de Ingeniería	
Número: 2	Número de Historia de Usuario: Funcionario, Administrador
Nombre Tarea: Ver Detalles de Personas con Registros Policiales	
Puntos estimados: 0.5	Tipo de tarea: Desarrollo
Descripción: Devuelve solo los datos que aparezcan en el sistema de la persona seleccionada.	

Tabla 6. HU 2 - Ver Detalles de Personas con Registros Policiales

Tarea de Ingeniería	
Número: 3	Número de Historia de Usuario: Funcionario, Administrador
Nombre Tarea: Consultar Vehículos con Registros Policiales.	
Puntos estimados: 1	Tipo de tarea: Desarrollo
Descripción: Permite al usuario consultar información operativa de vehículos con registros policiales dependiendo de los criterios de búsqueda especificados y se devuelve una lista de coincidencias de vehículos con registros policiales.	

Tabla 7. HU 3 - Consultar Vehículos con Registros Policiales.

Tarea de Ingeniería	
Número: 4	Número de Historia de Usuario: Funcionario, Administrador
Nombre Tarea: Ver Detalles de Vehículos con Registros Policiales.	
Puntos estimados: 0.5	Tipo de tarea: Desarrollo
Descripción: Devuelve solo los datos que aparezcan en el sistema del vehículo seleccionado.	

Tabla 8. HU 4 - Ver Detalles de Vehículos con Registros Policiales.

Tarea de Ingeniería	
Número: 5	Número de Historia de Usuario: Funcionario, Administrador
Nombre Tarea: Consultar Vehículos INTT.	
Puntos estimados: 1	Tipo de tarea: Desarrollo
Descripción: Permite al usuario consultar información operativa de vehículos INTT	

dependiendo de los criterios de búsqueda especificados y se devuelve una lista de coincidencias de vehículos con registros policiales.

Tabla 9. HU 5 - Consultar Vehículos INTT.

Tarea de Ingeniería	
Número: 6	Número de Historia de Usuario: Funcionario, Administrador
Nombre Tarea: Ver Detalles de Vehículos INTT.	
Puntos estimados: 0.5	Tipo de tarea: Desarrollo
Descripción: Devuelve solo los datos que aparezcan en el sistema del vehículo seleccionado.	

Tabla 10. HU 6 - Ver Detalles de Vehículos INTT.

Tarea de Ingeniería	
Número: 7	Número de Historia de Usuario: Funcionario, Administrador
Nombre Tarea: Consultar Armas con Registros Policiales.	
Puntos estimados: 1	Tipo de tarea: Desarrollo
Descripción: Permite al usuario consultar información operativa de armas con registros policiales dependiendo de los criterios de búsqueda especificados y se devuelve una lista de coincidencias de vehículos con registros policiales.	

Tabla 11. HU 7 - Consultar Armas con Registros Policiales.

Tarea de Ingeniería	
Número: 8	Número de Historia de Usuario: Funcionario, Administrador
Nombre Tarea: Ver Detalles de Armas con Registros Policiales.	
Puntos estimados: 0.5	Tipo de tarea: Desarrollo
Descripción: Devuelve solo los datos que aparezcan en el sistema del arma seleccionada.	

Tabla 12. HU 8 - Ver Detalles de Armas con Registros Policiales.

Tarea de Ingeniería	
Número: 9	Número de Historia de Usuario: Funcionario, Administrador
Nombre Tarea: Consultar Objetos con Registros Policiales.	

Tabla 13. HU 9 - Consultar Objetos con Registros Policiales.

Tarea de Ingeniería	
Número: 10	Número de Historia de Usuario: Funcionario, Administrador
Nombre Tarea: Ver Detalles de Objetos con Registros Policiales.	
Puntos estimados: 0.5	Tipo de tarea: Desarrollo
Descripción: Devuelve solo los datos que aparezcan en el sistema del objeto seleccionado.	

Tabla 14. HU 10 - Ver Detalles de Objetos con Registros Policiales.

Tarea de Ingeniería	
Número: 11	Número de Historia de Usuario: Funcionario, Administrador
Nombre Tarea: Consultar Persona Natural.	
Puntos estimados: 1	Tipo de tarea: Desarrollo
Descripción: Permite al usuario consultar información operativa de personas naturales dependiendo de los criterios de búsqueda especificados y se devuelve una lista de coincidencias de personas naturales.	

Tabla 15. HU 11 - Consultar Persona Natural.

Tarea de Ingeniería	
Número: 12	Número de Historia de Usuario: Funcionario, Administrador
Nombre Tarea: Ver Detalles de Persona Natural.	
Puntos estimados: 0.5	Tipo de tarea: Desarrollo
Descripción: Devuelve solo los datos que aparezcan en el sistema de la persona natural seleccionada.	

Tabla 16. HU 12 - Ver Detalles de Persona Natural.

Tarea de Ingeniería	
Número: 14	Número de Historia de Usuario: Administrador
Nombre Tarea: Crear Rol.	

Puntos estimados: 0.5	Tipo de tarea: Desarrollo
Descripción: Le permitirá al Administrador crear un rol.	

Tabla 17. HU 14 - Crear Rol.

Tarea de Ingeniería	
Número: 15	Número de Historia de Usuario: Administrador
Nombre Tarea: Modificar Rol.	
Puntos estimados: 0.5	Tipo de tarea: Desarrollo
Descripción: Le permitirá al Administrador modificar un rol determinado.	

Tabla 18. HU 15 - Modificar Rol.

Tarea de Ingeniería	
Número: 16	Número de Historia de Usuario: Administrador
Nombre Tarea: Eliminar Rol.	
Puntos estimados: 0.5	Tipo de tarea: Desarrollo
Descripción: Le permitirá al Administrador eliminar un rol.	

Tabla 19. HU 16 - Eliminar Rol.

Tarea de Ingeniería	
Número: 17	Número de Historia de Usuario: Administrador
Nombre Tarea: Crear Usuario	
Puntos estimados: 0.5	Tipo de tarea: Desarrollo
Descripción: Le permitirá al Administrador crear un usuario y asignarle los roles correspondiente.	

Tabla 20. HU 17 - Crear Usuario.

Tarea de Ingeniería

Número: 18	Número de Historia de Usuario: Administrador
Nombre Tarea: Modificar Usuario	
Puntos estimados: 0.5	Tipo de tarea: Desarrollo
Descripción: Le permitirá al Administrador modificar un usuario existente en el sistema, así como sus roles.	

Tabla 21. HU 18 - Modificar Usuario.

Tarea de Ingeniería	
Número: 19	Número de Historia de Usuario: Administrador
Nombre Tarea: Buscar Usuario	
Puntos estimados: 0.5	Tipo de tarea: Desarrollo
Descripción: Le permitirá al Administrador buscar un usuario existente en el sistema.	

Tabla 22. HU 19 - Buscar Usuario.

Tarea de Ingeniería	
Número: 20	Número de Historia de Usuario: Administrador
Nombre Tarea: Eliminar Usuario.	
Puntos estimados: 0.5	Tipo de tarea: Desarrollo
Descripción: Le permitirá al Administrador eliminar un usuario existente en el sistema.	

Tabla 23. HU 20 - Eliminar Usuario.

Tarea de Ingeniería	
Número: 21	Número de Historia de Usuario: Funcionario, Administrador
Nombre Tarea: Autenticar	
Puntos estimados: 0.5	Tipo de tarea: Desarrollo
Descripción: Le permitirá al usuario Autenticarse en la aplicación.	

Tabla 24. HU 21 – Autenticar.

ANEXO3: Tarjetas CRC

Tarjetas CRC para el Servidor.

Clase: ArmaDaoImpl	
Descripción: Clase encargada del acceso a datos de Arma.	
Responsabilidad	Colaborador
Consultar armas	
Ver datos de arma	

Clase: ObjetoDaoImpl	
Descripción: Clase encargada del acceso a datos de Objeto.	
Responsabilidad	Colaborador
Consultar objetos	
Ver datos de objeto	

Clase: VehiculoDaoImpl	
Descripción: Clase encargada del acceso a datos de Vehículo.	
Responsabilidad	Colaborador
Consultar vehículos	
Ver datos de vehículo	

Clase: VehiculoINTTDaoImpl	
Descripción: Clase encargada del acceso a datos de VehículoINTT.	
Responsabilidad	Colaborador
Consultar vehículos INTT	
Ver datos de vehículo INTT	

Clase: PersonaNaturalDaoImpl	
Descripción: Clase encargada del acceso a datos de Persona Natural.	

Responsabilidad	Colaborador
Consultar personas naturales	
Ver datos de persona natural	

Clase: PersonaDaolmpl	
Descripción: Clase encargada del acceso a datos de Persona.	
Responsabilidad	Colaborador
Consultar personas	
Ver datos de persona	

Clase: UsuarioDaolmpl	
Descripción: Clase encargada del acceso a datos de Usuario	
Responsabilidad	Colaborador
Consultar Usuario	
Consultar Funcionario CICPC	
Consultar Funcionario Externo	
Eliminar Usuario	
Verificar Usuario	
Obtener Usuario por Nombre	
Deshabilitar Usuario	

Clase: PerfilDaolmpl	
Descripción: Clase encargada del acceso a datos de Perfil	
Responsabilidad	Colaborador
Buscar Permisos	
Buscar Permisos por Perfil	
Consultar Perfil	

Buscar Perfiles por Usuario	
Eliminar Perfil	
Verificar Nombre de Perfil	

Clase: GenericDaolmpl	
Descripción: Clase genérica para el acceso a datos.	
Responsabilidad	Colaborador
Definir un comportamiento común para las clases de acceso a datos: -Salvar Entidad -Modificar Entidad -Eliminar Entidad -Ejecutar Procedimiento Almacenado	

Clase: ArmaServiceImpl	
Descripción: Clase encargada del negocio de Arma	
Responsabilidad	Colaborador
Consultar armas	ArmaDaolmpl
Ver datos de arma	TrazaServiceImpl

Clase: ObjetoServiceImpl	
Descripción: Clase encargada del negocio de Objeto.	
Responsabilidad	Colaborador
Consultar objetos	ObjetoDaolmpl
Ver datos de objeto	TrazaServiceImpl

Clase: VehiculoServiceImpl	
Descripción: Clase encargada del negocio de Vehículo.	
Responsabilidad	Colaborador
Consultar vehículos	VehiculoDaolmpl

Ver datos de vehículo	TrazaServiceImpl
-----------------------	------------------

Clase: VehiculoINTTServiceImpl	
Descripción: Clase encargada del negocio de VehículoINTT.	
Responsabilidad	Colaborador
Consultar vehículos INTT	VehiculoINTTDaoImpl
Ver datos de vehículo INTT	TrazaServiceImpl

Clase: PersonaNaturalServiceImpl	
Descripción: Clase encargada del negocio de Persona Natural.	
Responsabilidad	Colaborador
Consultar personas naturales	PersonaNaturalDaoImpl
Ver datos de persona natural	TrazaServiceImpl

Clase: PersonaServiceImpl	
Descripción: Clase encargada del negocio de Persona.	
Responsabilidad	Colaborador
Consultar personas	PersonaDaoImpl
Ver datos de persona	TrazaServiceImpl

Clase: UsuarioServiceImpl	
Descripción: Clase encargada del negocio de Usuario	
Responsabilidad	Colaborador
Consultar Usuario	UsuarioDaoImpl TrazaServiceImpl
Consultar Funcionario CICPC	
Consultar Funcionario Externo	
Eliminar Usuario	
Verificar Usuario	
Obtener Usuario por Nombre	

Deshabilitar Usuario	
----------------------	--

Clase: PerfilServiceImpl	
Descripción: Clase encargada del negocio de Perfil	
Responsabilidad	Colaborador
Buscar Permisos	PerfilDaoImpl TrazaServiceImpl
Buscar Permisos por Perfil	
Consultar Perfil	
Buscar Perfiles por Usuario	
Eliminar Perfil	
Verificar Nombre de Perfil	

Clase: TrazaServiceImpl	
Descripción: Clase encargada de las trazas	
Responsabilidad	Colaborador
Salvar las trazas de consulta de entidades operativas	TrazaDaoImpl
Salvar las trazas de gestión de usuarios y perfiles	

Clase: LoginService	
Descripción: Clase encargada de realizar el login	
Responsabilidad	Colaborador
Autenticarse	AuthenticationDaoImpl

Clase: LoginController	
Descripción: Clase encargada de manejar la petición de login	
Responsabilidad	Colaborador
Autenticarse	LoginService

Clase: ArmaControllerImpl	
----------------------------------	--

Descripción: Clase encargada de manejar las peticiones de Armas.	
Responsabilidad	Colaborador
Consultar armas	ArmaServiceImpl
Ver datos de arma	

Clase: ObjetoControllerImpl	
Descripción: Clase encargada de manejar las peticiones de Objetos.	
Responsabilidad	Colaborador
Consultar objetos	ObjetoServiceImpl
Ver datos de objeto	

Clase: PersonaControllerImpl	
Descripción: Clase encargada de manejar las peticiones de Personas.	
Responsabilidad	Colaborador
Consultar personas	PersonaServiceImpl
Ver datos de persona	

Clase: PersonaNaturalControllerImpl	
Descripción: Clase encargada de manejar las peticiones de Personas Naturales.	
Responsabilidad	Colaborador
Consultar personas naturales	PersonaNaturalServiceImpl
Ver datos de persona natural	

Clase: VehiculoControllerImpl	
Descripción: Clase encargada de manejar las peticiones de Vehículos	
Responsabilidad	Colaborador
Consultar vehículos	VehiculoServiceImpl
Ver datos de vehículo	

Clase: VehiculoINTTControllerImpl	
Descripción: Clase encargada de manejar las peticiones de Vehículos INTT	
Responsabilidad	Colaborador
Consultar vehículos INTT	VehiculoINTTServiceImpl
Ver datos de vehículo INTT	

Tarjetas CRC para el cliente.

Clase: Connection	
Descripción: Esta clase establece la conexión con el servicio web.	
Responsabilidad	Colaborador
Establece la conexión con el servicio web	DataParser

Clase:DataParser	
Descripción: Esta clase permite parsear la URL.	
Responsabilidad	Colaborador
Parsea la URL del servicio web	ParserAsyncTask

Clase: ParserAsyncTask	
Descripción: Clase encargada de los hilos de ejecución.	
Responsabilidad	Colaborador
Controlar todos los eventos en ejecución paralelos	DataParser

Clase: ConsultarObjeto

Descripción: Clase encargada de la consulta sobre los Objetos.	
Responsabilidad	Colaborador
Consultar objetos	ParserAsyncTask ResultadosObjeto

Clase: ConsultarVehiculoIntt	
Descripción: Clase encargada de la consulta sobre los Vehículos INTT.	
Responsabilidad	Colaborador
Consultar vehículos INTT	ParserAsyncTask ResultadosvehiculoIntt

Clase: ConsultarVehiculo	
Descripción: Clase encargada de la consulta de los Vehículos.	
Responsabilidad	Colaborador
Consultar vehículos	ParserAsyncTask ResultadosVehiculo

Clase: ConsultarPersonaSaime	
Descripción: Clase encargada de la consulta de las Personas Naturales.	
Responsabilidad	Colaborador
Consultar personas naturales	ParserAsyncTask

	ResultadosPersonaSaime
--	------------------------

Clase: ConsultarPersona	
Descripción: Clase encargada de la consulta de las Personas con delitos penales.	
Responsabilidad	Colaborador
Consultar personas	ParserAsyncTask ResultadosPersona

Clase: Login	
Descripción: Clase encargada autentificar al usuario.	
Responsabilidad	Colaborador
Comprobar rol del usuario y permitirle su acceso.	Consultar

Clase: Consultar	
Descripción: Clase encargada de permitir las opciones de consulta.	
Responsabilidad	Colaborador
Permitir las opciones de consulta	ConsultarPersona ConsultarPersonaSaime ConsultarVehiculo ConsultarVehiculoIntt

	ConsultarObjeto
	ConsultarArma

Clase: ResultadosPersona	
Descripción: Clase encargada de mostrar los datos de las Personas con delitos.	
Responsabilidad	Colaborador
Ver datos de persona	ConsultarPersona

Clase: ResultadosPersonaSaime	
Descripción: Clase encargada de mostrar los datos de las Personas naturales.	
Responsabilidad	Colaborador
Ver datos de personas naturales	ConsultarPersonaSaime

Clase: ResultadosVehiculo	
Descripción: Clase encargada de mostrar los datos de los vehiculos.	
Responsabilidad	Colaborador
Ver datos de los vehiculos	ConsultarVehiculo

Clase: ResultadosVehiculoIntt	
Descripción: Clase encargada de mostrar los datos de los vehículos INTT.	
Responsabilidad	Colaborador

Ver datos de los vehículos INTT	ConsultarVehiculoIntt
---------------------------------	------------------------------

Clase: ResultadosObjeto	
Descripción: Clase encargada de mostrar los datos de los objetos.	
Responsabilidad	Colaborador
Ver datos de los objetos	ConsultarObjetos

Clase: ResultadosArma	
Descripción: Clase encargada de mostrar los datos de las Armas.	
Responsabilidad	Colaborador
Ver datos de los Armas	ConsultarArma