

**DESARROLLO DE WEB SERVICES PARA PRESTAR LOS SERVICIOS
DE CORREO ELECTRÓNICO, PERFIL E INFORMACIÓN GENERAL
A LOS ESTUDIANTES DE LA FACULTAD DE INGENIERÍA
DE LA UNIVERSIDAD DE MANIZALES**

**CRISTIAN ANDRÉS VÁSQUEZ HENAO
LUKAS MAURICIO GÓMEZ DE LA PAVA**

**UNIVERSIDAD DE MANIZALES
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES
MANIZALES
2007**

**DESARROLLO DE WEB SERVICES PARA PRESTAR LOS SERVICIOS
DE CORREO ELECTRÓNICO, PERFIL E INFORMACIÓN GENERAL
A LOS ESTUDIANTES DE LA FACULTAD DE INGENIERÍA
DE LA UNIVERSIDAD DE MANIZALES**

**CRISTIAN ANDRÉS VÁSQUEZ HENAO
LUKAS MAURICIO GÓMEZ DE LA PAVA**

**Trabajo de Grado presentado como requisito parcial para optar
al título de Ingeniero de Sistemas y Telecomunicaciones**

**Presidente
LUIS MARCIAL BERTEL PATERNINA
Ingeniero Electricista**

**UNIVERSIDAD DE MANIZALES
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES
MANIZALES
2007**

CONTENIDO

	Pág.
INTRODUCCIÓN	10
1. DESCRIPCIÓN DEL ÁREA PROBLEMÁTICA	11
2. Objetivos	12
2.1 General	12
2.2 Específicos	12
3. Justificación	13
4. Marco Teórico	14
4.1 Herramientas de análisis	14
4.1.1 Unified Modeling Language, UML	14
4.2 Protocolos	15
4.2.1 General Packet Radio Services, GPRS	15
4.2.2 Internet Message Access Protocol, IMAP	16
4.2.3 Simple Message Transfer Protocol, SMTP	17
4.3 Otros	18
4.3.1 Servicio Web	18
4.3.2 Java Micro Edition, JME	20
4.3.3 Rich Site Summary, RSS	22
5. Metodología	24
5.1 Tipo de Trabajo	24
5.2 Procedimiento	24
5.2.1 Fase 1	24
5.2.2 Fase 2	24
5.2.3 Fase 3	25
5.2.4 Fase 4	25
5.2.5 Fase 5	25
6. Resultados	26
7. Conclusiones	28
8. Recomendaciones	29
Bibliografía	30
Anexos	33

LISTA DE ANEXOS

	Pág.
Anexo A. Análisis y Diseño del Módulo de noticias lector RSS	34
1 Análisis módulo lector de noticias RSSUM	35
1.1 Diagrama de clases	35
1.2 Diagrama de sucesos	36
1.3 Diagrama de colaboración	37
1.4 Diagrama de estados	38
1.5 Diagrama de flujo	39
1.5.1 Diagrama de flujo general	39
1.5.2 Diagrama de flujo para el proceso realizar conexión	39
1.5.3 Diagrama de flujo para el proceso analizar feed	40
1.6 Diagrama de casos de uso	41
1.6.1 Diagrama de casos de uso para la gestión de feeds	41
1.6.2 Diagrama de casos de uso para la conexión con la fuente RSS	42
1.7 Diagrama de componentes	43
1.8 Diagrama de despliegue	44
2 Diseño módulo lector de noticias RSSUM	45
2.1 Diagrama de clases	45
2.1.1 Diccionario de datos	47
2.2 Diagrama de secuencia	54
2.2.1 Diagrama de secuencia para la aplicación RSSUM	54
2.2.2 Diagrama de secuencia para la aplicación FeedEditor	58
2.3 Diagrama de colaboración	59
2.3.1 Diagrama de colaboración para la aplicación RSSUM	59
2.3.2 Diagrama de colaboración para la aplicación FeedEditor	60
2.4 Diagrama de despliegue	61
2.5 Diagrama de componentes	62
Anexo B. Análisis y Diseño del Módulo CorreoUM	63
1 Análisis módulo de correo CorreoUM	64
1.1 Diagrama de clases	64
1.2 Diagrama de sucesos	65
1.3 Diagrama de colaboración	66
1.4 Diagrama de estados	67
1.5 Diagrama de flujo	68
1.5.1 Diagrama de flujo general	68
1.5.2 Diagrama de flujo para el proceso efectuar conexión	69
1.5.3 Diagrama de flujo para el proceso gestión de mensajes	69
1.6 Diagrama de casos de uso	70
1.6.1 Diagrama de casos de uso para la conexión con el servidor de correo	70
1.6.2 Diagrama de casos de uso para la gestión de mensajes	71
1.7 Diagrama de componentes	73

1.8 Diagrama de despliegue	74
2 Diseño módulo correo	75
2.1 Diagrama de clases	76
2.1.1 Diccionario de datos	77
2.2 Diagrama de secuencia	81
2.3 Diagrama de colaboración	84
2.4 Diagrama de despliegue	85
2.5 Diagrama de componentes	86
Anexo C. Análisis y Diseño del Módulo de consulta ConsultaUM	87
1 Análisis módulo de consulta ConsultaUM	88
1.1 Diagrama de clases	88
1.2 Diagrama de sucesos	89
1.3 Diagrama de colaboración	90
1.4 Diagrama de estados	91
1.5 Diagrama de flujo	92
1.6 Diagrama de casos de uso	93
1.7 Diagrama de componentes	94
1.8 Diagrama de despliegue	95
2 Diseño módulo de consulta de notas ConsultaUM	96
2.1 Diagrama de clases	96
2.1.1 Diccionario de datos	97
2.2 Diagrama de secuencia	100
2.3 Diagrama de colaboración	101
2.4 Diagrama de despliegue	102
2.5 Diagrama de componentes	103
2.6 Diagrama WSDL para el servicio WEB	104
2.7 Diagrama de clases para el Servicio WEB	105
2.8 Diagrama de secuencia para el servicio WEB	105
Anexo D. Manual de usuario	106
Anexo E. Manual técnico	121
Anexo F. Documentos RFC para protocolos de correo	CD

LISTA DE FIGURAS

	Pág.
Figura 1. Página de descarga de aplicaciones	26
Figura 2. Aplicaciones	27

GLOSARIO

APLICACIÓN WEB: es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet. Las aplicaciones web son populares debido a la practicidad del navegador web. La facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad.

ASCII: acrónimo inglés de American Standard Code for Information Interchange, Código Estadounidense Estándar para el Intercambio de Información, es un código de caracteres basado en el alfabeto latino tal como se usa en inglés moderno y en otras lenguas occidentales. Fue creado en 1963 por el Comité Estadounidense de Estándares (ASA, conocido desde 1969 como el Instituto Estadounidense de Estándares Nacionales, o ANSI) como una refundición o evolución de los conjuntos de códigos utilizados entonces en telegrafía.

AUTENTICACION SMTP: configuración realizada con el fin de elevar los niveles de seguridad y eficacia del servicio de correo electrónico y con el objetivo de minimizar la posibilidad que la cuenta sea utilizada sin autorización, asimismo disminuirá la posibilidad que una cuenta sea utilizada como "puente" para el envío de correos masivos a través de servidores.

BASE DE DATOS: es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

CLIENTE: es una aplicación informática que se utiliza para acceder a los servicios que ofrece un servidor, normalmente a través de una red de telecomunicaciones.

DIRECCION IP: es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol).

E-BUSINESS: consiste principalmente en la distribución, compra, venta, mercado y suministro de información complementaria para productos o servicios a través de redes informáticas como Internet u otras.

INTERFAZ: parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario. Metafóricamente se entiende la interfaz como conversación entre el usuario y el sistema.

JAVA APPLETS: es un componente de una aplicación que corre en el contexto de otro programa, por ejemplo un navegador web. El applet debe correr en un contenedor, que lo proporciona un programa anfitrión, mediante un plugin, o en aplicaciones como teléfonos móviles que soportan el modelo de programación por applets.

LENGUAJES DE PROGRAMACION: es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente.

MEMORIA: se refiere a componentes de una computadora, dispositivos y medios de grabación que retienen datos informáticos durante algún intervalo de tiempo. Las memorias de computadora proporcionan una de las principales funciones de la computación moderna, la retención de información.

MODEM: es un equipo que sirve para modular y demodular (en amplitud, frecuencia, fase u otro sistema) una señal llamada portadora mediante otra señal de entrada llamada moduladora.

NAVEGADOR WEB: es una aplicación que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet. Cualquier navegador actual permite mostrar o ejecutar gráficos, secuencias de vídeo, sonido, animaciones y programas diversos además del texto y los hipervínculos o enlaces.

NODOS DE RED: Punto de intersección o unión de varios elementos que confluyen en el mismo lugar. Por ejemplo: En una red de ordenadores cada una de las máquinas es un nodo, y si la red es Internet, cada servidor constituye también un nodo.

PAGINA WEB: es una fuente de información adaptada para la World Wide Web (WWW) y accesible mediante un navegador de Internet. Ésta información se presenta generalmente en formato HTML y puede contener hiperenlaces a otras páginas web, constituyendo la red enlazada de la World Wide Web.

PDA: del inglés Personal Digital Assistant, (Ayudante personal digital) es un computador de mano originalmente diseñado como agenda electrónica (calendario, lista de contactos, bloc de notas y recordatorios) con un sistema de reconocimiento de escritura. Hoy día se puede usar como una computadora doméstica (ver películas, crear documentos, juegos, correo electrónico, navegar por Internet, escuchar música, etc.).

PORTAL WEB: sitio Web cuyo objetivo es ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse buscadores, foros, documentos, aplicaciones, compra electrónica, etc. Principalmente están dirigidos a resolver necesidades específicas de un grupo de personas o de acceso a la información y servicios de a una institución pública o privada.

PROCESADOR: Es el componente en una computadora digital que interpreta las instrucciones y procesa los datos contenidos en los programas de computadora.

PROCESOS (HILOS): Un hilo de ejecución es una característica que permite a una aplicación realizar varias tareas concurrentemente. Los distintos hilos de ejecución comparten una serie de recursos tales como el espacio de memoria, los archivos abiertos, situación de autenticación, etc. Esta técnica permite simplificar el diseño de una aplicación que debe llevar a cabo distintas funciones simultáneamente. Los hilos de ejecución que comparten los mismos recursos, sumados a estos recursos, son en conjunto conocidos como un proceso.

PROGRAMACION ORIENTADA A OBJETOS: La Programación Orientada a Objetos (POO u OOP según siglas en inglés) es un paradigma de programación (enfoque particular o filosofía para la construcción del software) que define los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (es decir, datos), comportamiento (esto es, procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas.

PROTOCOLO DE RED: es el conjunto de reglas que especifican el intercambio de datos u órdenes durante la comunicación entre las entidades que forman parte de una red.

PUERTO: forma genérica de denominar a una interfaz por la cual diferentes tipos de datos pueden ser enviados y recibidos. Dicha interfaz puede ser física, o puede ser a nivel software, por ejemplo: los puertos que permiten la transmisión de datos entre diferentes computadoras.

RED: es un conjunto de computadoras y/o dispositivos conectados por enlaces, a través de medios físicos (medios guiados) ó inalámbricos (medios no guiados) y que comparten información (archivos), recursos (CD-ROM, impresoras, etc.) y servicios (e-mail, chat, juegos), etc.

REDES INALÁMBRICAS: Una red inalámbrica es como cualquier otra red de computadores, conecta computadoras a redes de computadoras pero sin la necesidad de cables. Puede proveer acceso a otras computadoras, bases de datos, Internet, y en el caso de Wireless Lans, el hecho de no tener cables, les permite a los usuarios contar con movilidad sin perder la conexión.

FEEDS RSS: Se refiere a un tipo de dato empleado para suministrar información que es actualizada con frecuencia. Se emplea para denominar a los documentos con formato RSS o Atom, basados en XML, que permiten a los agregadores recoger información de páginas web sindicadas.

SERVIDOR WEB: programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

SERVIDOR: una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de un ordenador y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final. Este es el significado original del término. Es posible que un ordenador cumpla simultáneamente las funciones de cliente y de servidor.

SOFTWARE: son todos los componentes intangibles de una computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware). Esto incluye aplicaciones informáticas tales como un procesador de textos, que permite al usuario realizar una tarea, y software de sistema como un sistema operativo, que permite al resto de programas funcionar adecuadamente, facilitando la interacción con los componentes físicos y el resto de aplicaciones.

WEB: es un medio de comunicación de texto, gráficos y otros objetos multimedia a través de Internet, es decir, la web es un sistema de hipertexto que utiliza Internet como su mecanismo de transporte o desde otro punto de vista, una forma gráfica de explorar Internet.

RESUMEN

En el desarrollo de este proyecto se implementaron tres aplicaciones, con el fin de permitir a los usuarios de dispositivos móviles de la Facultad de Ingeniería de la Universidad de Manizales, acceder a una parte de la información que ofrece el portal Athenea para los estudiantes de ingeniería, como las notas de las asignaturas, las faltas de asistencia, las noticias de la Facultad y correo electrónico.

Para el desarrollo del proyecto se utilizaron distintos recursos que permitieron crear aplicaciones ligeras y eficientes. Enfocados en un análisis y diseño orientado a objetos según los criterios del Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language), y haciendo uso del lenguaje de programación Java Micro Edition (JME) en el ambiente de desarrollo integrado (IDE, por sus siglas en inglés, Integrated Development Environment) NEtBeans 5.0 con la extensión Mobility Pack 5.0 RC 2, se desarrollaron cada una de las aplicaciones.

El módulo lector de noticias se desarrolló para Sindicación Realmente Simple (RSS, por sus siglas en inglés, Really Simple Syndication), que toma un archivo en el Lenguaje de Etiquetas Extensible (XML, por sus siglas en inglés, Extensible Markup Language) el cual pasa a través de una interfaz para programación de aplicaciones (API, por sus siglas en inglés, Application Programming Interface), analizador XML (XML Parser), que se encarga de analizar las etiquetas para extraer la información útil.

En el módulo de correo electrónico se desarrolló un cliente que se vale del Protocolo Simple de Transferencia de Mensajes (SMTP, por sus siglas en inglés, Simple Mail Transfer Protocol) para el envío de correo y del Protocolo de Acceso a Mensajes de Internet (IMAP, por sus siglas en inglés, Internet Message Access Protocol) para la lectura de correo de Athenea; este cliente de correo se hizo usando sockets, que según el libro *Sistemas Operativos*¹, se define como un punto final para comunicación. Un socket está compuesto de una dirección IP concatenada con un número de puerto. Un par de procesos (o hilos) que se comunican por una red emplean un par de sockets, uno para cada proceso; para leer comandos y respuestas de los protocolos, con el fin de gestionar el correo de los usuarios.

¹ SILBERSCHATZ, Abraham; GALVIN, Peter y GAGNE, Greg. *Sistemas Operativos*. 6 ed. México D.F.: Editorial Limusa, S.A. de C.V. 2002. 839 p. ISBN: 968-18-6168-X.

El módulo de consulta de notas y asistencia se divide en dos partes: la primera parte es un servicio Web que se encuentra en el servidor GLEC y realiza la conexión con la base de datos y la consulta; la segunda parte es una aplicación Web que funciona como cliente, usando el servicio Web para obtener los datos desde el manejador de base de datos Postgres de Athenea; las notas y asistencia mostradas en este módulo corresponden a las asignaturas inscritas por cada estudiante en el semestre en curso.

Por último, se desarrolló una página Web que permite descargar las aplicaciones al dispositivo móvil, para hacer uso de ellas inmediatamente, sin necesidad de conectarse a un portal cada vez que se haga una consulta.

ABSTRACT

Developing this Project, were applied three different applications focused in led the mobile applications users of the Manizales University (Engineer Program) to access part of the information who has Athenea Site for the engineer students, as grades, assistant control tabs, program news, and e-mail.

To develop this project were used different resources to create light and efficient applications, focused in the analyze and design of objects according to the Unified Modeling Language (UML), and using the Java Micro Edition (JME) in the Integrated development Environment (IDE) Netbeans 5.0 with the extension Mobility Pack 5.0 RC 2.

The lector Module was development to the Really Simple Syndication (RSS); it takes an Extensible Markup Language (XML) file, which goes throw the Applications Programming Interface (API), XML parser who has in charge to analyze the labels to extract important information.

On the e-mail module was development a client that uses Simple Mail Protocol (SMTP) to send the e-mail and the internet message access protocol (IMAP). To read the Athenea E-mails this user was created using sockets. A socket is composed of an IP address with a port number and defined as a critical point to communication. A pair of process that are communicated by a network uses a pair of sockets, one by process; to read commands and answer protocols, with the idea of move the users e-mails.

Checking grades and assistance module is divided in two parts: the first one is a web service that is located in the GLEC Server and realize the connection between the data base and does the query; the second part is a web application that Works as a client, using the web service to get the dates from Postgres data base manager; the grades and the assistance showed in this module are corresponding to the signed signatures by each student of the semester in course.

The last challenge was to develop a web site to download the applications to the mobile item, to use them immediately, without a site connection for searching purpose.

INTRODUCCIÓN

Dada la masificación del uso de dispositivos móviles y la capacidad de los mismos para acceder a redes de comunicaciones utilizando distintas técnicas, se abren espacios para la generación de plataformas que permitan la conexión a estos dispositivos, brindando a sus usuarios una gama casi ilimitada de información, que sólo se verá restringida por las necesidades que tengan las organizaciones que implementen dichas plataformas. De esta manera, instituciones como la Universidad de Manizales, pueden aprovechar estos avances tecnológicos para ofrecer a los estudiantes, información importante sobre su proceso académico.

En la solución informática desarrollada para ampliar el acceso a la información mencionada, se utiliza el Servicio General de Paquetes de Radio (GPRS, por sus siglas en inglés, General Packet Radio Service) empleado por las compañías de telefonía celular del país para la transmisión de datos de dispositivos móviles a través de Internet, con lo que se consigue el acceso de los usuarios de éstos dispositivos a las aplicaciones desarrolladas en este proyecto.

Estas aplicaciones hacen uso del Protocolo Simple de Transferencia de Mensajes (SMTP, por sus siglas en inglés, Simple Mail Transfer Protocol) y del Protocolo de Acceso a Mensajes de Internet (IMAP, por sus siglas en inglés, Internet Message Access Protocol), que permiten respectivamente enviar y leer los correos electrónicos de las cuentas de cada usuario, y del formato de noticias Sindicación Realmente Simple (RSS, por sus siglas en inglés, Really Simple Syndication) que facilita la extracción de información contenida en archivos en Lenguaje de Etiquetas Extensible (XML, por sus siglas en inglés, Extensible Markup Language); los recursos mencionados han sido elegidos por compatibilidad con el servidor Athenea, puesto que las aplicaciones desarrolladas hacen uso de sus servicios, a diferencia de la aplicación de consulta de notas, que emplea un servicio Web encargado de obtener la información de la base de datos, montado en el servidor GLEC para el uso específico de dicha aplicación.

Con el uso de los distintos protocolos y herramientas de desarrollo, se logra una solución que permite un acceso inmediato a la información, limitado por la disponibilidad y área de cobertura de la red celular de la compañía a la cual el usuario se encuentre suscrito.

1. DESCRIPCIÓN DEL ÁREA PROBLEMÁTICA

Las instituciones académicas deben ofrecer mayor disponibilidad de información a sus estudiantes en los temas concernientes a su proceso formativo y otros aspectos de orden institucional que le sea útil a éstos, y para alcanzar este objetivo, deben utilizar recursos que permitan hacer una difusión ágil y eficiente de dicha información.

Actualmente, el uso de la telefonía celular se masifica a un ritmo vertiginoso, lo que ha motivado a que los productores de servicios de comunicación trabajen decididamente en el desarrollo de nuevas tecnologías y herramientas, con el propósito de mejorar la experiencia de los usuarios al acceder a servicios de datos a través de la red de telefonía celular.

Aunque a partir del año 2006 en la Facultad de Ingeniería de la Universidad de Manizales se introdujo, en las asignaturas afines a la programación, el desarrollo de aplicaciones para dispositivos móviles; no se puede desconocer que, hasta hace pocos años no había en Colombia un enfoque directo hacia el desarrollo de este tipo de aplicaciones, no existían en las universidades asignaturas en las que se dieran a conocer las herramientas y/o la teoría para su implementación, razón por la cual la comunidad académica de ingeniería carecía de interés por esta área. Teniendo en cuenta que la mayoría de los estudiantes de la Facultad de Ingeniería de la Universidad de Manizales tienen un dispositivo móvil, se hace evidente la existencia de un recurso valioso, que estaba siendo desaprovechado y del cual la Facultad puede hacer uso para ofrecer información a los estudiantes.

La información en este momento se ofrece a los estudiantes de la Facultad de Ingeniería a través del servidor académico Athenea, éste consta de un portal web, un servidor de correo electrónico, secciones de consulta de notas, noticias, contactos, búsquedas, encuestas, información académica y personal de docentes, estudiantes y administrativos; pero éste se encuentra sujeto a la necesidad de disponer de un computador personal para su uso, limitando el acceso.

2. OBJETIVOS

2.1 GENERAL

Diseñar e implementar Web Services para la Facultad de Ingeniería de la Universidad de Manizales, conectados al servidor académico Athenea, que permitan a los usuarios de dispositivos móviles de la Universidad acceder a la información inherente a su proceso académico (correo, notas, asistencia, noticias), sin estar sujetos a un espacio y tiempo determinados.

2.2 ESPECÍFICOS

- Diseñar aplicaciones haciendo uso de Web Services que permita acceder a la información académica de la Universidad de Manizales por medio del servidor Athenea, desde dispositivos móviles.
- Crear una Interfaz Gráfica de Usuario (GUI, por sus siglas en inglés Graphic User Interface) amigable, que permita a los usuarios revisar la información de una manera rápida, sencilla y segura
- Brindar acceso a través de las aplicaciones desarrolladas, a toda la información requerida por los usuarios (estudiantes de la Facultad de Ingeniería), para ofrecer un servicio integral (notas, correo, noticias).
- Desarrollar un software escalable de tal manera que pueda garantizar la agregación de otros módulos en el futuro, como puede ser el financiero o datos personales.

3. JUSTIFICACIÓN

La Facultad de Ingeniería de la Universidad de Manizales, para ser una facultad líder en los niveles regional, nacional e internacional, debe mantenerse a la vanguardia en cuanto a tecnología, y el desarrollo de aplicaciones como Web Services, es un paso más para lograr este objetivo y convertirse en una de las pocas facultades en Colombia, que implementa una tecnología capaz de ofrecer a los estudiantes una gran variedad de información y servicios útiles por medio de los dispositivos móviles, lo que debe permitir que los estudiantes puedan conocer la información inherente a su proceso formativo en cualquier momento y desde cualquier lugar.

Aunque el Campus Virtual de la Universidad de Manizales desarrolló un servicio de consulta de notas dentro de su página WAP, éste proyecto ofrece el servicio de consulta de notas utilizando servicios web, lo que no solo permite mayor compatibilidad a futuro, ya que los dispositivos móviles ya están contando con el soporte para Web; si no que en el momento que se desee migrar a plataforma Web el portal WAP, se puede hacer uso del servicio para consultar los datos. Además, el proyecto cuenta también con los servicios de noticias y correo electrónico de Athenea.

Este desarrollo satisface la necesidad de los estudiantes de tener acceso oportuno a la información, y la necesidad de la Facultad, de ofrecer mayor variedad de herramientas de calidad para acompañar el proceso académico de sus estudiantes.

4. MARCO TEÓRICO

4.1 HERRAMIENTAS DE ANÁLISIS

4.1.1 UNIFIED MODELING LANGUAGE, UML. Rumbaugh, Booch y Grady², expresan que el lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es un lenguaje de modelado visual de propósito general usado para especificar, visualizar, construir y documentar el comportamiento de un sistema, para ello captura decisiones y comportamientos del mismo que deben ser construidos con el fin de entender, diseñar, explorar, configurar, mantener y controlar información del sistema. Fue desarrollado para usarse con todos los métodos de desarrollo, niveles del ciclo de vida, dominios de aplicación y medios; también para unificar experiencias pasadas sobre técnicas de modelado e incorporar las mejores prácticas de software actuales en una propuesta estándar. UML incluye conceptos semánticos, notación y directrices, tanto como componentes estáticos, dinámicos, de entorno y organizacionales con el fin de facilitar el proceso de modelado. Puede ser soportado por herramientas interactivas de modelado visual que contienen generadores de código y escritores de reportes. La especificación UML no define un proceso estándar pero puede usarse con un desarrollo de procesos iterativo y permite su implementación en la mayoría de procesos de desarrollo orientado a objetos.

La captura de información sobre la estructura estática y dinámica del comportamiento de un sistema, el cual es modelado como una colección de objetos discretos que interactúan para realizar un trabajo que beneficia a un usuario exterior, es la base para la aplicación de UML. La estructura estática define la clase de objetos importantes para un sistema y su implementación, así como las relaciones entre estos objetos. El comportamiento dinámico define el historial de objetos sobre el tiempo y las comunicaciones entre los objetos para cumplir una tarea. El modelado de un sistema desde estas perspectivas con puntos de vista relacionados, permite entender los diferentes propósitos de éste.

UML contiene constructores organizacionales para arreglos de modelos en paquetes, que permiten dividir sistemas robustos en piezas trabajables para su desarrollo, con el fin de entender y controlar las dependencias entre paquetes y manejar las versiones de unidades de modelado en un entorno de desarrollo complejo; además permite representar decisiones de implementación y organizar elementos de tiempo de ejecución en componentes.

Aunque algunas herramientas pueden proveer generadores de código desde UML hacia una variedad de lenguajes de programación, así como construir modelos por

² RUMBAUGH, James; JACOBSON, Ivar y BOOCH, Grady. UML Reference Manual. Reading Massachusetts: Addison Wesley Longman, Inc., 1999. 568 p. ISBN 0-201-30998-X.

medio de ingeniería inversa a partir de programas existentes, éste no es un lenguaje de programación.

4.2 PROTOCOLOS

4.2.1 GENERAL PACKET RADIO SERVICES, GPRS. En la Guía Completa de Protocolos de Comunicaciones³, se afirma que GPRS se usa como una actualización para servicios de datos de cualquier red GSM (Sistema Móvil Global, por sus siglas en inglés, Global System Mobile); permite que las redes GSM sean realmente compatibles con internet. GPRS utiliza una técnica en modo paquete para transferir tráfico a ráfagas de manera eficiente. Permite velocidades de bit de transmisión desde 9.6 Kbps hasta más de 150 Kbps por usuario.

Las dos ventajas del servicio GPRS son un mejor uso de los recursos de radio y de red, y que soporta IP de forma completamente transparente. GPRS optimiza el uso de los recursos de red y de radio: solo utiliza los recursos de radio cuando hay datos que hay que enviar o recibir. Como auténtica tecnología de paquetes, permite que las aplicaciones de usuario final solo ocupen la red cuando se está transfiriendo carga útil y está, por tanto, bien adaptado a la naturaleza de tipo ráfaga de las aplicaciones de datos.

Otra importante característica del servicio GPRS es que proporciona conectividad inmediata y una alta tasa de transferencia. GPRS soporta aplicaciones basadas en protocolos de datos estándar como IP y X.25. En GPRS hay disponibles cuatro niveles de calidad de servicio diferentes. Para dar soporte a aplicaciones de datos, GPRS utiliza varios nuevos nodos de red. Estos nodos son responsables del encaminamiento del tráfico y de otras funciones de interoperación con redes externas de datos de conmutación de paquetes; también se encarga de la ubicación de suscriptores, selección de celdas, itinerancia y muchas otras funciones que cualquier red celular necesita para funcionar.

4.2.2 INTERNET MESSAGE ACCESS PROTOCOL, IMAP. Como se encuentra en el libro Redes de Computadoras de Tanenbaum⁴, IMAP, a diferencia de POP3, que asume básicamente que el usuario vaciará el buzón de cada contacto y trabajará sin conexión después de eso, IMAP supone que todo el correo electrónico permanecerá en el servidor de manera indefinida en múltiples buzones de correo. IMAP proporciona mecanismos de gran alcance para leer mensajes e incluso partes de un mensaje, una característica útil cuando se utiliza un modem

³ MCGRAW-HILL. Serie de Telecomunicaciones: Guía Completa de Protocolos de Telecomunicaciones. Madrid: McGraw-Hill/Interamericana de España, S.A.U. 2002. 607 p. ISBN: 84-481-3641-1.

⁴ TANENBAUM, S. Andrew. Redes de Computadoras. 4 ed. México: Pearson Education de México, S.A. 2003. 891 p. ISBN: 970-26-0162-2

lento para leer parte del texto de un mensaje dividido en varios fragmentos y que tiene archivos adjuntos grandes de audio y video. Debido a que la suposición más razonable es que los mensajes no se transferirán a la computadora del usuario para un almacenamiento permanente, IMAP proporciona mecanismos para crear destruir y manipular múltiples buzones en el servidor. De esta forma, un usuario puede mantener un buzón para cada uno de sus contactos y colocar ahí mensajes de la bandeja de entrada después de que se ha leído.

IMAP tiene muchas características como la capacidad de dirigir correo no por número de llegada, sino utilizando atributos como por ejemplo el primer mensaje de un contacto determinado. A diferencia de POP3, IMAP también puede aceptar correo saliente para enviarlo al destino, así como entregar correo electrónico entrante.

El estilo general del protocolo IMAP es similar al de POP3, excepto porque POP3 posee docenas de comandos. El servidor IMAP escucha en el puerto 143. Se debe mencionar que no todos los proveedores de servicios de internet ni todos los programas de correo electrónico soportan ambos protocolos.

Los documentos de tipo Petición de Comentarios (RFC, por sus siglas en inglés, Request for Comment) del protocolo IMAP contienen detalles y especificaciones sobre cómo debe funcionar el protocolo. El documento RFC-1730 define en primer lugar el modo en el que el protocolo IMAP debe utilizar la versión 4, pero RFC-2060 contiene las cuestiones de implantación de IMAP actuales que utilizan muchos servidores IMAP y que se denomina versión IMAP4rev1.

4.2.3 SIMPLE MESSAGE TRANSFER PROTOCOL, SMTP. Como se expone en el libro Redes de Computadoras de Tanenbaum⁵ SMTP es un protocolo ASCII sencillo. Después de establecer la conexión TCP con el puerto 25, la maquina emisora, operando como cliente, espera que la maquina receptora, operando como servidor, hable primero. El servidor comienza enviando una línea de texto que proporciona su identidad e indica si está preparado o no para recibir correo. Si no lo está, el cliente libera la conexión y lo intenta después.

Si el servidor está dispuesto a aceptar correo electrónico el cliente anuncia de quien proviene el mensaje, y a quien va dirigido. Si existe el destinatario en el destino, el servidor da al cliente el permiso para enviar el mensaje. A continuación el cliente envía el mensaje y el servidor confirma su recepción. Por lo general, no se requieren sumas de verificación porque TCP proporciona un flujo de bytes confiable. Si hay mas correo electrónico, se envía ahora. Una vez que todo el correo electrónico ha sido intercambiado en ambas direcciones, se libera la conexión.

⁵ TANENBAUM, S. Andrew. Redes de Computadoras. 4 ed. México: Pearson Education de México, S.A. 2003. 891 p. ISBN: 970-26-0162-2

El primer comando del cliente es HELO. El siguiente es RCPT que indica a quien se mandará el mensaje, se permiten varios comandos RCPT para enviar el mensaje a correos múltiples; se confirma la recepción de cada uno o se rechaza de manera individual. Incluso si se rechazan algunos destinatarios, el mensaje puede enviarse a los demás.

El comando FROM se utiliza luego de RCPT para identificar la cuenta de correo desde la que se envía el mensaje. Seguido de este, el comando DATA en el que se escribe el mensaje que se enviará terminando este con un punto (.) en una línea vacía.

En el documento RFC-2821 se describe el comportamiento básico del protocolo SMTP, aunque varias extensiones de SMTP, posibles gracias a RFC-1869, han agregado nuevas funciones al SMTP a lo largo de los años con nuevos comandos. Al iniciar una conversación con un servidor SMTP mediante un comando EHLO, en lugar de HELO, el servidor conectado puede identificarse a sí mismo como un servidor compatible con las extensiones SMTP. El servidor receptor contesta con una línea 250 que contiene las distintas extensiones SMTP compatibles. A continuación, el servidor conectado puede utilizar las extensiones compatibles como desee para obtener los objetivos de la comunicación.

Una de las extensiones que merece la pena destacar está relacionada con la incorporación de autenticación SMTP mediante el comando AUTH, tal y como se describe en el documento RFC-2554. Otra extensión SMTP muy utilizada se explica en detalle en el documento RFC-2034, que describe el uso entre aplicaciones SMTP de códigos de error estándar separados por puntos. La lectura de los documentos RFC en los que se describen aspectos del protocolo SMTP proporciona los conocimientos básicos sobre la forma de transferir el correo electrónico en Internet. Además, puede conectarse a un servidor SMTP mediante telnet si especifica el puerto 25, como, por ejemplo telnet localhost 25. Ejecute varios comandos y envíe correos electrónicos manualmente para aprender cómo funcionan las comunicaciones SMTP.

4.3 OTROS

4.3.1 SERVICIO WEB. Según Liliana M. Arboleda⁶, los servicios web son aplicaciones auto-contenidas, auto-descriptivas y modulares, que pueden ser publicadas, localizadas e invocadas a través del web y que cuentan con un mecanismo estándar para establecer la comunicación con otros tipos de software a través de la red.

⁶ ARBOLEDA C., Liliana M. Servicios Web: Distribución e Integración [En línea]. Santiago de Cali, Octubre 2004. Disponible en internet: http://dspace.icesi.edu.co/dspace/bitstream/item/403/1/larboled_servicios-web.pdf .

Los servicios web permiten integrar las aplicaciones de una manera eficiente, sin preocuparse por cómo fueron construidas, donde residen, sobre qué sistema operativo se ejecutan o cómo acceder a ellas. Precisamente por esta razón es que los servicios web se consideran integradores, porque permiten crear una interfaz de acceso a las aplicaciones, sin importar las características de implementación de éstas. Previamente, para hacer este tipo de integración, era necesario crear módulos de software especializados, pero actualmente con los servicios web, básicamente se trabaja con estructuras basadas en XML.

Un servicio web simple está caracterizado por cuatro estándares: XML, SOAP, UDDI y WSDL, los cuales al trabajar juntos proporcionan una funcionalidad básica de tipo “solicitud/ respuesta”. Los servicios web simples pueden utilizarse para entregar de forma eficiente información como noticias, inventarios y reportes de clima a los sitios web, mientras que un servicio web complejo puede involucrar transacciones más elaboradas entre varias partes, involucrando socios de negocios o proveedores y basado en los estándares de e-business (por ej.: ebXML).

Desde un nivel superior, un servicio web se puede definir como una unidad de código que puede ser activada utilizando solicitudes http. Históricamente hablando, el acceso remoto a unidades binarias requiere protocolos específicos de cada plataforma (y algunas veces específicos de cada lenguaje).

Utilizando servicios web, un programador puede emplear cualquier lenguaje que desee, y un consumidor de servicios web puede usar http estándar para invocar métodos en los tipos definidos en el servicio web, la única restricción real a la que debe ponerse atención es que debido a que los servicios web son diseñados para facilitar las invocaciones remotas, debe evitarse la utilización de cualquier lógica basada en la interfaz gráfica de usuario (GUI).

CARACTERÍSTICAS Y OPERACIONES DE LOS SERVICIOS WEB

- Elementos fundamentales para el esquema de funcionamiento de los servicios web:

1. Un proveedor del servicio web, que es quien lo diseña, desarrolla e implementa y lo pone disponible para su uso.
2. Un consumidor del servicio, que es quien accede al componente para utilizar los servicios que éste presta.
3. Un agente de servicio, que sirve como enlace entre proveedor y consumidor para efectos de publicación, búsqueda y localización del servicio.

- Definición.

Al definir un servicio web, se requiere contar con lo siguiente:

a) Para colocar disponible el servicio:

- El primer paso es definir el servicio web: como se dijo antes, el componente software puede desarrollarse utilizando diferentes lenguajes y tecnologías, pero para que este componente pueda interactuar con otros, es necesario utilizar un lenguaje común que permita estructurar los datos que componen el servicio web. En este caso tal lenguaje es XML (eXtended Markup Language).
- Una vez se ha definido la funcionalidad del servicio, es necesario publicarlo para que otros servicios y aplicaciones puedan acceder a él. Las operaciones de publicación involucran el anuncio del servicio como tal, lo cual corresponde a la ubicación del servicio en un servidor específico y el uso de un servicio de descripción.

El servicio de descripción se utiliza para que los clientes puedan saber qué funciones tiene disponibles el servicio web y qué información debe pasarse a esas funciones para poder utilizarlas, y para esto el consumidor debe entender totalmente los miembros expuestos. Por ejemplo, el consumidor debe saber que el servicio ofrece un método llamado primer Método (string parámetro1, boolean parámetro2, int parámetro3) que toma tres parámetros de tipo string, boolean e int respectivamente, y que devuelve un tipo llamado Respuesta, antes de poder invocarlo. El lenguaje utilizado para elaborar esta descripción del servicio web se denomina WSDL (Web Service Description Language) y la publicación se hace mediante UDDI (Universal Description, Discovery and Integration).

b) Para localizar el servicio:

- Cuando un consumidor desea acceder a un servicio web, debe contar con un servicio de descubrimiento, que permita conocer la ubicación exacta del servicio, es decir, se debe contar con un directorio donde se tengan listas las referencias a los servicios disponibles. Esto se logra gracias a UDDI.

Para intercambiar información hacia y desde el servicio (entre los consumidores y los servicios web), es necesario contar con un protocolo de comunicaciones como SOAP, el cual trasmite los datos sobre HTTP, utilizando métodos como GET o POST. La información entre un servicio web y su consumidor se transmite en formato XML.

Estándares primarios de un Servicio Web

- XML: eXtensible Markup Language

Permite representar documentos -documento se refiere no sólo a textos, sino también gráficos, correos electrónicos, transacciones de comercio electrónico, etc.- a través de una estructura lógica y una física. Físicamente, el documento está

compuesto de unidades llamadas entidades. Una entidad, en términos más simples, puede verse como un trozo de documento, es decir, puede ser un simple carácter o todo el capítulo de un libro. Todo documento tiene una entidad raíz.

En un documento XML es posible hacer referencia a entidades internas o externas, indicando que están definidas en el mismo documento o en otros documentos -que pueden estar en otros archivos del disco duro, en una base de datos, en internet, etc.-, lo que hace muy flexible la construcción de documentos XML. Para definir un elemento en XML se tienen etiquetas y contenido, que en XML se denominan datos de tipo carácter. Las estructuras, física y lógica de un documento se organizan adecuadamente, para lo cual se han establecido unas normas que definen cuándo un documento XML está bien formado y cuándo es válido, o no.

- WSDL: Web Services Definition Language

Lenguaje común utilizado para la descripción de los servicios web basado en XML, que describe totalmente la forma en la cual los clientes externos pueden interactuar con los servicios web existentes en una máquina dada, los métodos que soportan y la sintaxis de los protocolos de comunicación (http, SOAP).

En términos generales, un documento WSDL contiene información acerca de la interfaz, la semántica y los aspectos administrativos involucrados en una solicitud (llamado) a un servicio web. Las operaciones y mensajes se describen en forma abstracta y luego se asocian a un protocolo de red o a un formato de mensaje específico, para definir un punto final. WSDL se extiende para permitir la descripción de los puntos finales y sus mensajes, sin importar qué formatos de mensaje o protocolos de red se estén utilizando para la comunicación.

- SOAP: Simple Object Access Protocol

Protocolo liviano basado en XML, utilizado para el intercambio de información estructurada en un ambiente descentralizado y distribuido. SOAP no define la aplicación, ni la semántica de implementación; en vez de esto, proporciona un modelo de empaquetamiento modular y los mecanismos para la codificación de los datos dentro de los módulos. Esto permite que el protocolo simple se utilice en una amplia variedad de sistemas modulares y que cumpla su propósito primario, el paso de parámetros y comandos entre clientes y servidores de http, independientemente de las plataformas y aplicaciones existentes en el cliente y en el servidor. Los parámetros y los comandos se codifican utilizando XML.

- UDDI: Universal Discovery, description and integration

Para los servicios web, XML es el mecanismo por medio del cual los servicios y/o clientes acuerdan lo que van a hablar unos a otros, SOAP indica cómo van a efectuar la comunicación y UDDI les permite conocer con quién se van a comunicar y dónde encontrar otros objetos. En pocas palabras, UDDI es una

especificación para un registro distribuido de información acerca de los servicios web y define la forma en la cual se publica y descubre información acerca de éstos. En un registro basado en UDDI se pueden descubrir los servicios web; el método utilizado por UDDI para el descubrimiento de servicios es tener un registro de aquellos servicios que se encuentran distribuidos a través del web, en el registro distribuido, los negocios y los servicios se describen utilizando un formato XML común. Los datos estructurados en esos documentos XML son de fácil búsqueda, análisis y manipulación.

Como su nombre lo indica, el estándar de descripción universal, descubrimiento e integración (UDDI), proporciona un mecanismo para que los servicios Web se describan a sí mismos y a los tipos de servicios que proporcionan, para luego registrarse y publicarse en un registro UDDI.

4.3.2 JAVA MICRO EDITION, JME. Según está asentado en el portal educativo español Ciberaula⁷, al principio de los 90, Sun Microsystems creó un nuevo lenguaje de programación llamado Oak como parte de un proyecto de investigación para construir productos electrónicos que dependan principalmente del software. El primer prototipo para Oak fue un controlador portable llamado Star7, un pequeño dispositivo con una pantalla touchscreen LCD que tenía incorporado soporte a redes inalámbricas y comunicaciones a través de infrarrojo. Este dispositivo podría ser usado como control remoto para televisores o VCR y como guía de programas electrónicos, e incluso tenía algunas funciones que ahora son asociadas a los PDAs, como agenda de citas. El software para este tipo de dispositivos necesitaba ser extremadamente confiable y no debía hacer excesivo uso de memoria ni requerir demasiada potencia en el procesador. Oak fue desarrollado como resultado de la experiencia del equipo de desarrollo con el lenguaje C++, el cual, a pesar de tener muchas grandes características, demostró que era un lenguaje complejo y ocasionaba que los programadores comentan fácilmente errores y eso afectaba la confiabilidad del software.

Oak fue diseñado para quitar o reducir la posibilidad de que los programadores comentan errores, ¿cómo? detectando la mayoría de errores en tiempo de compilación y quitando algunas de las características del lenguaje C++ (como punteros y la administración de memoria controlada por el programador) que eran los problemas más comunes.

Desafortunadamente, el mercado para el tipo de dispositivos que el nuevo lenguaje fue creado no se desarrolló tanto como Sun Microsystems esperaba, y al final ningún dispositivo basado en Oak fue vendido a los clientes. Sin embargo, al mismo tiempo, el inicio del conocimiento público de Internet produjo un mercado para el software de navegación para Internet (los navegadores Web). En respuesta a esto, Sun Microsystems renombró el lenguaje de programación Oak a

⁷ CIBERAULA. Introducción al J2ME. [En línea]. Madrid (España): Ciberaula, Fecha de actualización: 2006. Disponible en Internet: http://java.ciberaula.com/articulo/introduccion_j2me/

Java y lo usó para desarrollar un navegador multiplataforma llamado HotJava. También le dio la licencia de Java a Netscape, quienes lo incorporaron en su navegador que por ese entonces era el más popular en el mercado, luego fueron incorporados los Java applets.

En un par de años, las capacidades multiplataforma del lenguaje de programación Java y su potencia como plataforma de desarrollo para aplicaciones que podían ser escritas una vez y ejecutadas en diversos sistemas Windows y Unix, había despertado el interés de usuarios finales, porque vieron en ella una manera de reducir los costos del desarrollo de software.

Con el objetivo de conocer las necesidades de los experimentados desarrolladores en Windows y Motif/X-Windows para crear aplicaciones para usuarios finales sofisticados acostumbrados a usar interfaces ricas, Sun Microsystems rápidamente expandió el alcance y tamaño de la plataforma Java. Esta plataforma extendida incluyó un conjunto más complejo de librerías y de interfaces de usuario que se usan para construir applets, además con un conjunto de características de computación distribuida y seguridad mejorada.

Con el tiempo Sun Microsystems liberó la primera versión de la plataforma Java 2, había sido necesario dividirla en varias piezas. La funcionalidad principal, estimado como el mínimo soporte requerido para cualquier ambiente Java, estaba empaquetada en el Java 2 Standard Edition (J2SE).

Muchos paquetes opcionales pueden ser agregados al J2SE para satisfacer requerimientos específicos para aplicaciones particulares, como extensiones seguras de sockets que permitan el comercio electrónico. Sun Microsystems también respondió al incremento del interés de usar Java para el desarrollo a un nivel empresarial, y ambientes de servidores de aplicaciones con la plataforma Java 2 Enterprise Edition (J2EE), el cual incorpora nuevas tecnologías como servlets, Enterprise JavaBeans, JavaServer pages, etc.

Como la mayoría de software, los requerimientos de recursos de Java tienen un incremento con cada nueva versión que aparece. A pesar que Java tiene sus raíces en el software para productos electrónicos pequeños, J2SE requiere mucha más memoria y potencia en el procesador para que sea una solución viable en el mercado.

Irónicamente, mientras Sun Microsystems estaba desarrollando Java para Internet y para la programación comercial, la demanda empezó a crecer en los dispositivos pequeños e incluso en tarjetas inteligentes, retornando Java a sus raíces. Sun Microsystems respondió a esta demanda creando varias plataformas Java con funcionalidades reducidas, cada una hecha a la medida de un segmento vertical y específico del mercado.

Estas plataformas reducidas están todas basadas en el JDK 1.1, el predecesor de la plataforma Java 2, y cada una tiene una estrategia diferente para abordar el

problema de reducir la plataforma para acomodarla a los recursos disponibles. Por lo tanto, cada una de estas plataformas de funcionalidad reducida, representan una solución de acuerdo al problema. Por ello es que aparece la plataforma JME, para reemplazar todas esas plataformas reducidas basadas en el JDK 1.1 y crear una sola solución basada en Java.

En resumen, JME es la versión de Java orientada a los dispositivos móviles. Debido a que los dispositivos móviles tienen una potencia de cálculo baja e interfaces de usuario pobres, es necesaria una versión específica de Java destinada a estos dispositivos, ya que el resto de versiones de Java, JSE o JEE, no encajan dentro de este esquema. JME es por tanto, una versión reducida de JSE.

4.3.3 RICH SITE SUMMARY, RSS. Como expone Cervantes⁸, RSS es un formato estandarizado para compartir encabezados y/o descripciones completas de las notas de periódicos en línea, portales, anuncios clasificados o cualquier otra información disponible en un sitio Web. RSS es una forma de sindicación para contenidos publicados en la Web; entendiéndose sindicación como la presencia del contenido de algún medio de comunicación en otros medios del mismo o diferente tipo.

RSS surgió como una iniciativa de UserLand, una compañía de software, pero fue utilizada por Netscape para dividir la información de su portal en canales como economía, tecnología, espectáculos, etcétera. A través de los RSS, los usuarios podían escoger la información que querían que apareciera en su visión personalizada del portal (My Netscape).

RSS 1.0, conocido como Rich Site Summary, surge de una implementación de la Estructura de Descripción de Recursos (RDF, por sus siglas en inglés, Resource Description Framework), un ligero derivado de XML, utilizado para describir e intercambiar metadatos de los datos que hay en la Web como: la bibliografía de un documento en una biblioteca digital o los datos más importantes de un sitio, que pueden ser de utilidad para los motores de búsqueda.

En RSS, el archivo que tiene los encabezados y cualquier otro dato que se desee compartir se llama RSS feed; además este archivo se publica en Internet y se genera de forma dinámica para que la información que se distribuya siempre sea la más reciente.

RSS para editores de sitios Web

⁸ CERVANTES GONZÁLEZ, Germán I. RSS, ahora la información que te interesa te encuentra a ti. [En línea]. México: Universidad Nacional Autónoma de México, Dirección General de Servicios de Cómputo Académico, publicación mensual: Entérate en línea, Fecha de Actualización: 04.2007. Disponible en internet: <http://www.enterate.unam.mx/Articulos/2005/febrero/rss.htm>

Los editores de un sitio deciden qué van a difundir a través de RSS, ya sea: los encabezados más recientes de un blog, de un periódico en línea, de una revista o de las noticias de un portal, las 10 aplicaciones más descargadas o el menú de un sitio. Los sitios Web que alberguen esta información no tendrán que modificar nunca alguna línea del contenido importado, ya que a medida que se actualice el RSS feed del sitio fuente, los destinos se actualizarán de forma automática.

RSS para lectores de sitios Web

En los sitios Web que han agregado RSS feeds en sus contenidos, los visitantes se topan con los encabezados, o con un resumen corto de los artículos o información del sitio Web original, los cuales son enlaces que los llevarán a la información completa que les llamó la atención.

Además de que los RSS feeds pueden ser agregados como parte del contenido de un sitio, existe una gran variedad de clientes o programas, llamados RSS readers o RSS aggregators, que permiten leer RSS feeds, y entre sus características pueden tener las de interpretar varias versiones de RSS feeds, organizarlos en una taxonomía jerárquica de acuerdo con diferentes categorías, ordenar los encabezados por diferentes atributos como: fecha de alta, título, fecha de vencimiento y la característica más importante, avisar en cuanto hay una modificación del RSS feed, mostrando de inmediato los nuevos encabezados y permitiendo a los lectores estar al tanto de las últimas noticias, en el momento que se publican.

Así, muchas prácticas como el envío de boletines vía e-mail, servicios de avisos, noticias o la actualización de contenidos redundantes en diferentes sitios, pueden ser automatizadas con el uso de RSS.

RSS 2.0, bautizado como Really Simple Syndication, en realidad sí es más simple, como su nombre lo dice, ya que omite las especificaciones de RDF, que podían parecer más complicadas, y especifica un documento derivado directamente de XML.

5. METODOLOGÍA

5.1 TIPO DE TRABAJO

Este proyecto corresponde a un desarrollo tecnológico, debido a que no se realiza ninguna investigación en estricto sentido y se utilizan herramientas y metodologías ya existentes para la creación de aplicaciones, lo que constituye un desarrollo lateral de una plataforma tecnológica ya consolidada.

5.2 PROCEDIMIENTO

5.2.1 Fase 1. Análisis y diseño del proyecto, utilizando las técnicas de modelado UML con el fin de enfocar el desarrollo hacia la programación orientada a objetos.

Esta fase comprendió las siguientes actividades:

- Elaborar los diagramas de clases, sucesos, colaboración, estados, flujo, casos de uso, componentes y despliegue (Análisis).
- Elaborar los diagramas de clases, secuencia, colaboración, despliegue y componentes, basados en el análisis realizado (Diseño).
- Seleccionar las herramientas con las que se desarrollan los módulos.

5.2.2 Fase 2. Programación, en JME, de las aplicaciones de correo electrónico, noticias por sindicación RSS y consulta de notas y asistencia. JME permite el desarrollo de aplicaciones para dispositivos móviles y actualmente es soportado por la mayoría de ellos.

Esta fase comprendió las siguientes actividades:

- Desarrollo del módulo correo electrónico: en este módulo se utilizan los protocolos estándar SMTP e IMAP, descritos en los numerales 4.2.2 y 4.2.3; se usaron sockets para leer línea por línea, cadenas de texto que representan los comandos y respuestas de estos protocolos.
- Desarrollo del módulo lector de noticias: este módulo se basa en los ejemplos de Sun Microsystems sobre lectura de RSS y se hace uso de la api *kXML12Parser* de la autoría de Jonathan Knudsen, que permite extraer de las etiquetas en el archivo XML, la información útil. La descripción de esta clase, llamada *rss.kxml12parser.java*, se puede ver en el anexo A, literal 2.1.1 (Diccionario de datos), página 53. En este módulo se extrae la

información de las etiquetas de un archivo en formato XML, de una fuente de datos RSS que se encuentra en Internet.

- Desarrollo del módulo consulta de notas y asistencia: se trabaja con Web services, pues permite una ágil y muy compatible transmisión de datos. El servicio se incorpora en el servidor GLEC y espera la llamada de un cliente móvil desde el que un usuario ingresa los datos que servirán para que el servicio realice la consulta a la base de datos de Athenea, para devolverle al cliente una cadena de texto con toda la información de sus notas y asistencia, para ser mostrada al usuario.

Las clases desarrolladas en estos módulos y su comportamiento, se encuentran detalladas en el análisis y diseño, en los anexos A, B y C.

5.2.3 Fase 3. Elaboración de la página web para descargar las aplicaciones, en el Lenguaje de Etiquetas de Hipertexto (HTML, por sus siglas en inglés, Hypertext Markup Language).

Esta fase comprendió las siguientes actividades:

- Desarrollo de la página web para descargar las aplicaciones mediante HTML: se creó una página sencilla que solo contiene los enlaces para descargar las aplicaciones.

5.2.4 Fase 4. Incorporación y configuración de la página de descarga de aplicaciones y el servicio Web en el servidor GLEC (implantación).

El servidor GLEC cuenta con el sistema operativo Ubuntu 7.0.4, se montó el servidor Web Sun Java System Application Server 8.2, pues tiene soporte completo para Web Services. En cuanto al manejo de bases de datos, se consulta sobre la base de datos que se encuentra en el manejador de bases de datos PostgreSQL de Athenea, para lo cual se debió agregar el driver para la conexión con PostgreSQL llamado postgresql-8.1-408.jdbc3.jar en el JDK instalado en el servidor.

Esta fase comprendió las siguientes actividades:

- Incorporación de la página de descarga de aplicaciones en el servidor Web de GLEC.
- Incorporación del servicio Web de consulta de notas en el servidor Web de GLEC.
- Configuración del manejador de base de datos de Athenea para aceptar conexiones del servicio Web de consulta de notas y asistencia.

Para ver en detalle estas configuraciones, referirse al Anexo E (Manual Técnico)

5.2.5 Fase 5. Desarrollo de pruebas de usabilidad, este tipo de prueba se refiere al correcto funcionamiento, y a asegurar que la interfaz de usuario (GUI) sea intuitiva y amigable.

Esta fase comprendió las siguientes actividades:

- Prueba de usabilidad al módulo CorreoUM.
- Prueba de usabilidad al módulo RssUM.
- Prueba de usabilidad al módulo ConsultaUM.
- Prueba de usabilidad de la página de descargas.

Estas pruebas se ejecutaron a medida que se desarrollaban los módulos; cuando se terminaba cada módulo, se revisaba en busca de errores, cuando se encontraba algún error o inconsistencia se volvía al proceso hasta corregir los errores, para de nuevo remitirse al proceso de pruebas.

6. RESULTADOS

El resultado del trabajo realizado son tres aplicaciones con las que se consultan diferentes elementos informativos que son útiles para los estudiantes de la Facultad de Ingeniería, y una página Web de la que pueden ser descargadas estas aplicaciones.

El usuario debe acceder a una dirección de Internet, mediante el navegador de su móvil, en la que se encuentra la página Web, la cual presenta las aplicaciones en una interfaz sencilla y ágil que mediante enlaces, permite descargarlas al dispositivo móvil, de manera que el usuario no tiene que volver a ingresar a ninguna página Web, pues ya cuenta con las herramientas.

Estas aplicaciones dependen del servidor Athenea para obtener la información aunque son independientes entre sí; esta independencia provee gran escalabilidad al desarrollo pues pueden agregarse fácilmente otras aplicaciones con diferentes objetivos, poseen una GUI simple pero amigable al igual que la página Web y funcionan como se explicará a continuación: La aplicación CorreoUM accede a la cuenta de correo electrónico que es asignada por la Facultad a los estudiantes mediante el servidor de correo de Athenea. CorreoUM permite leer los mensajes que llegan a la bandeja de entrada de la cuenta y redactar mensajes para ser enviados a distintos destinatarios siempre y cuando el servidor de correo maneje autenticación SMTP.



Figura 1. Página de descarga de aplicaciones

RssUM es un lector de noticias RSS que permite la lectura de noticias que presenta Athenea; además permite la suscripción a distintas páginas Web que ofrecen este tipo de noticias. RssUM toma los archivos XML y los analiza para extraer la información de cada noticia y mostrarla en la pantalla del móvil.

NotasUM es una aplicación que accede a un servicio Web que se incorpora al servidor GLEC y que realiza una consulta a la base de datos de Athenea para conseguir los datos de las asignaturas matriculadas por cada estudiante. NotasUM recibe la información que ha obtenido el servicio Web y la organiza de modo que el usuario pueda ver las calificaciones de cada corte por asignatura, con sus respectivas faltas de asistencia. Las notas que presenta la aplicación corresponden a las notas digitadas en Athenea durante el semestre que se esté cursando.

Para el desarrollo del proyecto se encontraron esencialmente dos inconvenientes, ambos relacionados con el módulo de correo; el primero se debe a que se trabajó con sockets, que según el libro *Sistemas Operativos*⁹, se define como un punto final para comunicación. Un socket está compuesto de una dirección IP concatenada con un número de puerto. Un par de procesos (o hilos) que se comunican por una red emplean un par de sockets, uno para cada proceso; lo que implicó realizar todas las funciones al nivel de los protocolos, lo que se tradujo en más tiempo de desarrollo, ya que el lenguaje JME no soporta algunas interfaces para programación de aplicaciones (API, por sus siglas en inglés, Application Programming Interface) de Java; como la API javamail, que contiene las funciones básicas de gestión de correo y es una de las más utilizadas para este tipo de aplicaciones.

La segunda, se debe a una limitación del servidor de correo de Athenea, pues no maneja autenticación SMTP, factor indispensable para el envío de correo desde fuera de la red local, lo que impide el envío de correo desde los móviles.



Figura 2. Aplicaciones

⁹ SILBERSCHATZ, Abraham; GALVIN, Peter y GAGNE, Greg. *Sistemas Operativos*. 6 ed. México D.F.: Editorial Limusa, S.A. de C.V. 2002. 839 p. ISBN: 968-18-6168-X.

7. CONCLUSIONES

Haciendo una revisión del proyecto, se puede concluir que:

- El proyecto genera un producto útil para otras instituciones educativas, debido a sus funciones interesantes y a su fácil implementación en distintos entornos con modificaciones simples. El desarrollo de la solución y su implantación en distintos entornos no conlleva un gasto significativo para una institución académica y resulta una herramienta práctica para la difusión de información.
- Los servicios Web permiten el desarrollo de soluciones de manera sencilla y práctica, generando servicios que pueden ser accedidos por diferentes aplicaciones. Se pueden crear operaciones pequeñas pero útiles para problemas específicos, que pueden ser empleados por diferentes usuarios.
- Según las pruebas realizadas a las aplicaciones, estas cumplen con el objetivo establecido, pues son de fácil manejo para cualquier usuario con experiencia en utilización de dispositivos móviles; además son rápidas, usables y arrojan resultados satisfactorios en una interfaz simple y completa.
- El montaje de servicios Web en el servidor SJSAS es muy sencillo y eficiente, gracias a que este posee un soporte completo para este tipo de servicios a diferencia de otros servidores más populares como el TOMCAT, que no ha sido desarrollado para este propósito lo que requiere complementos como el axis e instalación de librerías, que por problemas de compatibilidad de versiones ocupan demasiado tiempo de implantación para una tarea sencilla.

8. RECOMENDACIONES

- Incluir en el servidor de correo de Athenea, la autenticación SMTP para enviar correo desde el cliente móvil, puesto que la aplicación está preparada para ello.
- Ampliar el rango de codificación en el módulo de correo electrónico, ya que existen gran variedad de ellas, y aunque los clientes de correo solo suelen dar soporte a utf-8 y latin, que son las más utilizadas, lo que haría que el servicio fuera más completo.
- Extender el servicio de consulta de notas, con el fin de permitir la consulta de notas de todos los semestres cursados por el estudiante.
- Dar soporte MIME al cliente de correo, para habilitar el manejo de archivos adjuntos.
- Para un futuro mejoramiento del modulo de correo, se puede añadir el soporte para conexiones seguras tanto en IMAP como para SMTP.

BIBLIOGRAFÍA

CHATTERJEE, Sandeep y WEBBER, James. Developing Enterprise Web Services: An Architect's Guide. New Jersey: Prentice Hall PTR Pearson Education, Inc., 2003. 592 p. ISBN: 0-13-140160-2.

JUNTAO YUAN, Michael. Enterprise J2ME: Developing Mobile Java Applications. New Jersey: Prentice Hall Professional Technical Reference. 2003. 480 p. ISBN: 0-13-140530-6.

MCGRAW-HILL. Serie de Telecomunicaciones: Guía Completa de Protocolos de Telecomunicaciones. Madrid: McGraw-Hill/Interamericana de España, S.A.U. 2002. 607 p. ISBN: 84-481-3641-1.

RUMBAUGH, James; JACOBSON, Ivar y BOOCH, Grady. UML Reference Manual. Reading Massachusetts: Addison Wesley Longman, Inc., 1999. 568 p. ISBN 0-201-30998-X.

SILBERSCHATZ, Abraham; GALVIN, Peter y GAGNE, Greg. Sistemas Operativos. 6 ed. México D.F.: Editorial Limusa, S.A. de C.V. 2002. 839 p. ISBN: 968-18-6168-X

TANENBAUM, S. Andrew. Redes de Computadoras. 4 ed. México D.F.: Pearson Education de México, S.A. 2003. 891 p. ISBN: 970-26-0162-2

Fuentes en línea:

ARBOLEDA C., Liliana M. Servicios Web: Distribución e Integración [En línea]. Santiago de Cali, Octubre 2004. Disponible en internet: http://dspace.icesi.edu.co/dspace/bitstream/item/403/1/larboled_servicios-web.pdf

CIBERAULA. Introducción al J2ME. [En línea]. Madrid (España): Ciberaula, Fecha de Actualización: 2006. Disponible en Internet: http://java.ciberaula.com/articulo/introduccion_j2me/

CERVANTES GONZÁLEZ, Germán I. RSS, ahora la información que te interesa te encuentra a ti.[En línea]. México: Universidad Nacional Autónoma de México,

Dirección General de Servicios de Cómputo Académico, publicación mensual: Entérate en línea, Fecha de Actualización: 04.2007. Disponible en internet: <http://www.enterate.unam.mx/Articulos/2005/febrero/rss.htm>

Fuentes adicionales:

- Desarrollo:

W3C. Web Services Architecture: W3C Working Draft 14 November 2002. [En línea]. Michael Champion, Software AG. Chris Ferris, IBM. Eric Newcomer, Iona. David Orchard, BEA Systems, Fecha de Actualización: 2002. Disponible en internet: <http://www.w3.org/TR/2002/WD-ws-arch-20021114/>.

NETBEANS. Tech Tip: Deploy and Run Java ME Applications using Sun Java Application Server. [En Línea]. Brisbane, California. CollabNet, Inc., Fecha de Actualización: 2007. Disponible en internet: <http://www.netbeans.org/kb/50/j2me-app-server.html>.

NETBEANS. Netbeans Mobility Pack 5.0 Quick Start Guide. [En Línea]. Brisbane, California. CollabNet, Inc., Fecha de Actualización: 2007. Disponible en internet: <http://www.netbeans.org/kb/50/quickstart-mobility.html>.

NETBEANS. NetBeans IDE 5.0 Quick Start Guide for Web Services. [En Línea]. Brisbane, California. CollabNet, Inc., Fecha de Actualización: 2007. Disponible en internet: <http://www.netbeans.org/kb/50/quickstart-webservice.html>.

NETBEANS. NetBeans IDE 5.0 Quick Start Guide for Web Services Clients. [En Línea]. Brisbane, California. CollabNet, Inc., Fecha de Actualización: 2007. Disponible en internet: www.netbeans.org/kb/50/quickstart-webservice-client.html.

NETBEANS. NetBeans IDE 5.0 Tutorial for Web Services Client. [En Línea]. Brisbane, California. CollabNet, Inc., Fecha de Actualización: 2007. Disponible en internet: www.netbeans.org/kb/50/tutorial-webservice-client.html.

- Antecedentes:

CAMPUSWAP.COM: Primer portal universitario de telefonía móvil. [En línea]. Vicerrectorado de Metodología, Medios y Tecnología. Campuswap.com. Primer portal universitario de telefonía móvil. Disponible en internet: <http://campuswap.com/index.html>. 2000.

CIUDAD INTERNET. [En línea]. Grupo Clarin, Fecha de Actualización: 2005. Ciudad Internet. Disponible en internet: <http://wap.ciudad.com.ar/>.

GOOGLE. Google Colombia. [En línea]. Google, Fecha de Actualización: 2005. Google Colombia. Disponible en internet: <http://wap.google.com>

HAZAR. Hazar. [En línea]. [H@z@r.com](http://www.hazar.com), Fecha de Actualización: 2005. Disponible en internet: <http://www.hazar.com/dictionary.wnl>

INFOBOLSA. Infobolsa. [En línea]. Infobolsa S.A., Fecha de Actualización: 2005. Infobolsa. España. Disponible en internet: <http://wap.infobolsa.es>

INFOCLIMA. [En línea]. Todo Clima SRL., Fecha de consulta 14.09.05. Infoclima. Disponible en internet: [http:// www.infoclima.com/wap/](http://www.infoclima.com/wap/)

LA NACION. La Nación Line. [En línea]. La Nación S.A., Fecha de Actualización: 2005. La Nación Line. Disponible en internet: <http://www.lanacion.com.ar/herramientas/wap/>

LA REVISTA UTEQ. Revista Universitaria. [En línea]. Universidad Técnica Estatal de Quevedo, Fecha de Consulta 14.09.05. Quevedo, Ecuador. La Revista UTEQ. Revista Universitaria. Disponible en internet: http://www.uteq.edu.ec/index2/revista/revista_noticia.php?recordID=25#

YAHOO. [En línea]. Yahoo inc., Fecha de Actualización: 2005. Yahoo. Disponible en internet: <http://es.wap.yahoo.com/>

ZEDGE. Zedge: free ringtones, free wallpapers, free themes, free downloads to Nokia and other mobile phones. [En línea]. Zedge Ltd., Fecha de Actualización: 2004. Zedge: free ringtones, free wallpapers, free themes, free downloads to Nokia and other mobile phones. Disponible en internet: <http://wap.zedge.no>

ANEXOS

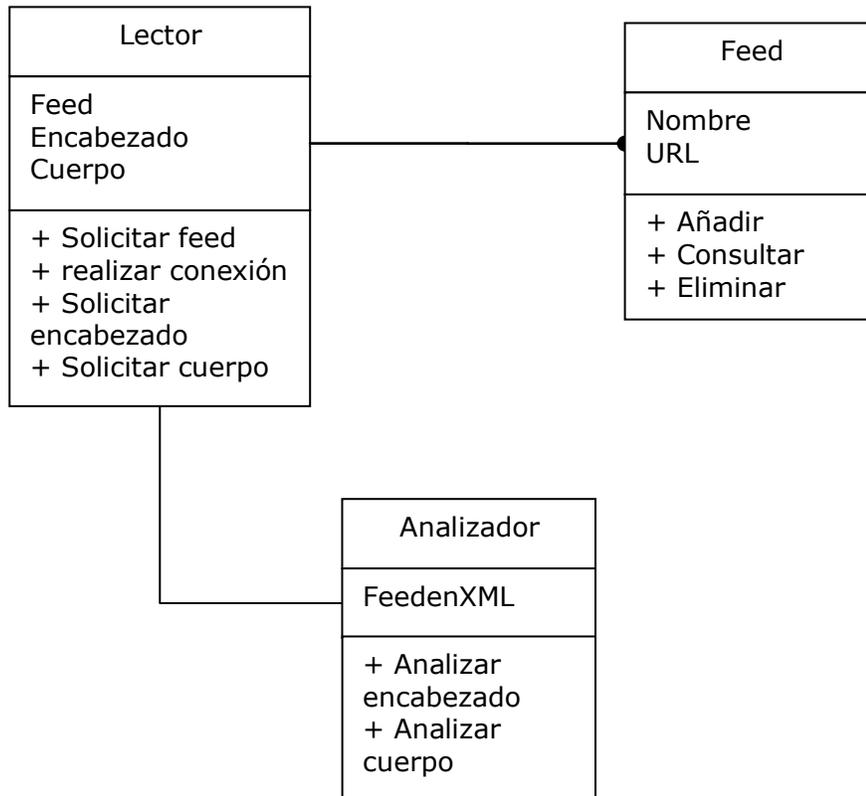
ANEXO A

ANÁLISIS Y DISEÑO DEL MÓDULO DE NOTICIAS LECTOR RSS

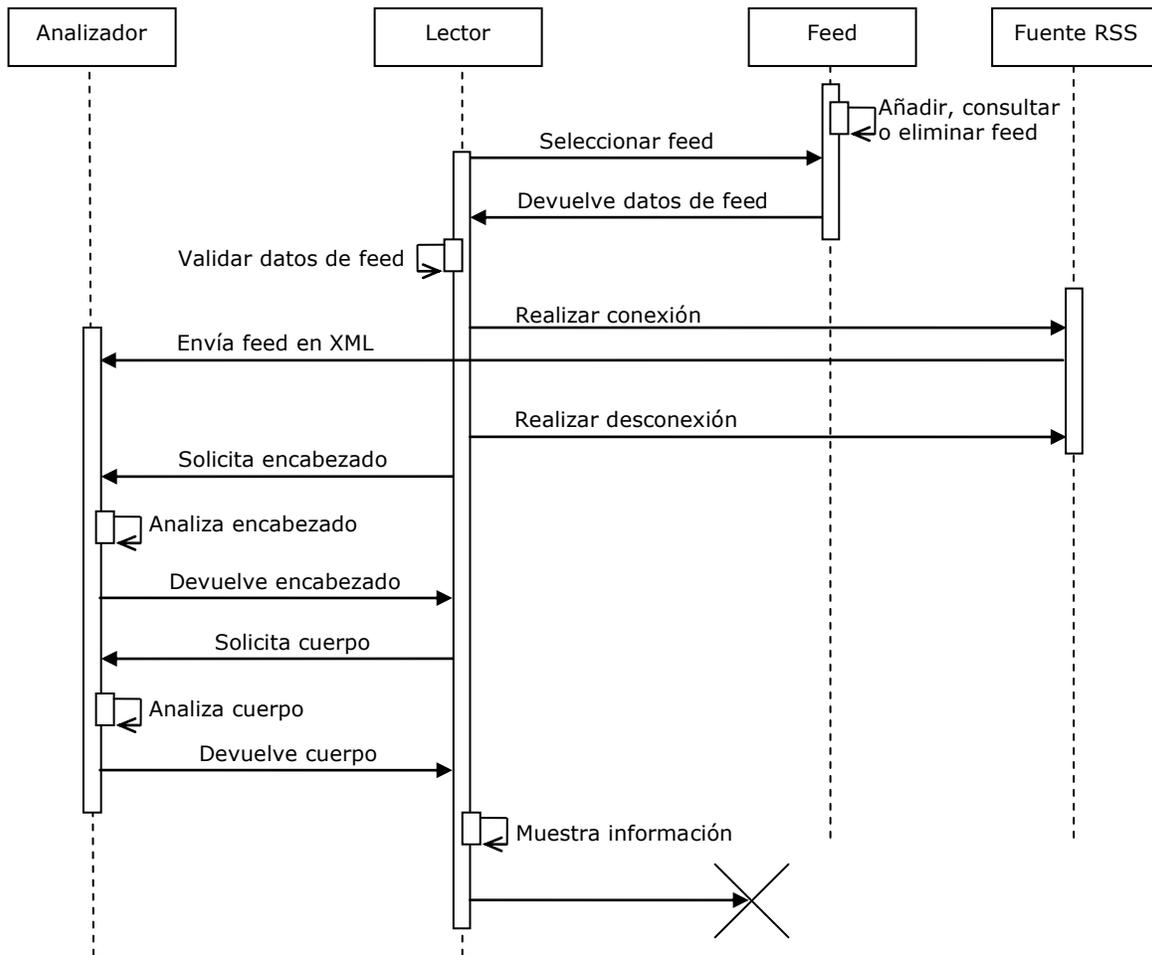
Descripción: este módulo es un cliente RSS, su función consiste en descargar las noticias en el formato RSS del portal Athenea, extraer la información y desplegarla en el dispositivo móvil, además permite anexar o eliminar feed RSS de otros sitios Web.

1. Análisis Módulo Lector de Noticias RSSUM

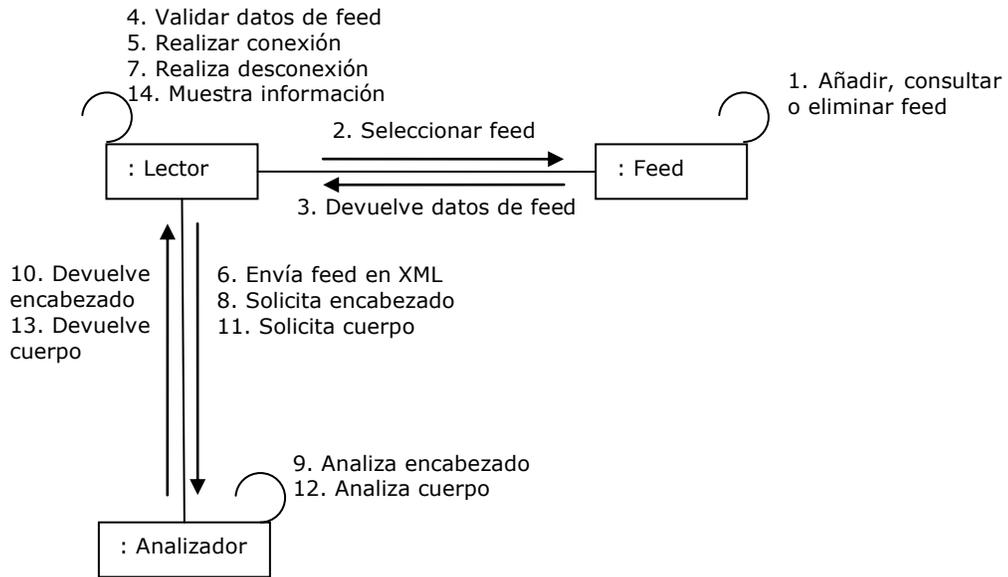
1.1. Diagrama de clases



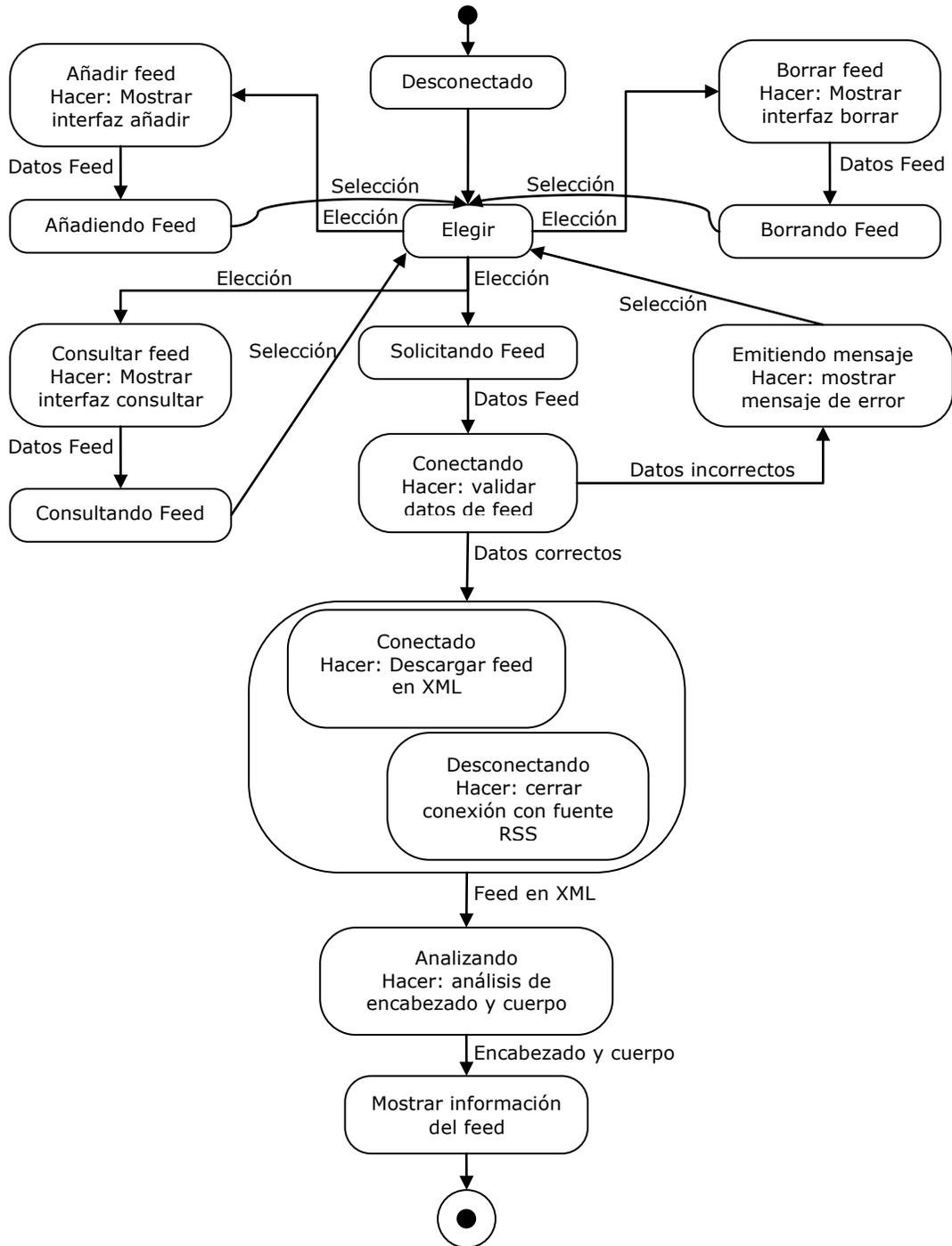
1.2. Diagrama de sucesos



1.3. Diagrama de colaboración

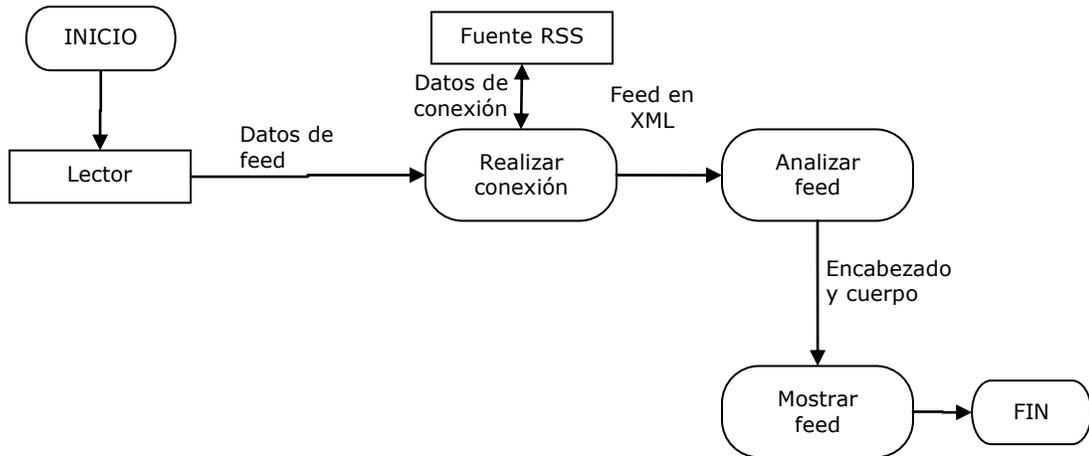


1.4. Diagrama de estados

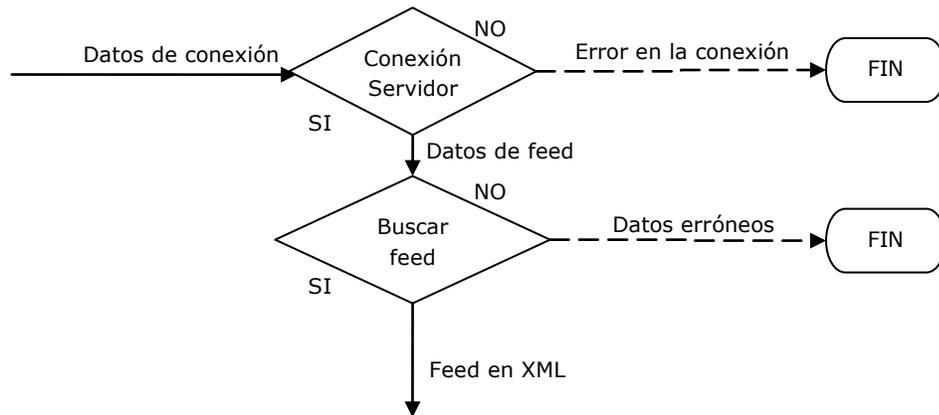


1.5. Diagrama de flujo

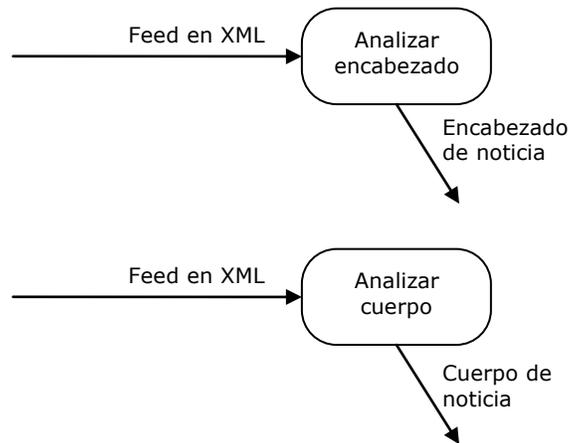
1.5.1. Diagrama de flujo general



1.5.2. Diagrama de flujo para el proceso realizar conexión

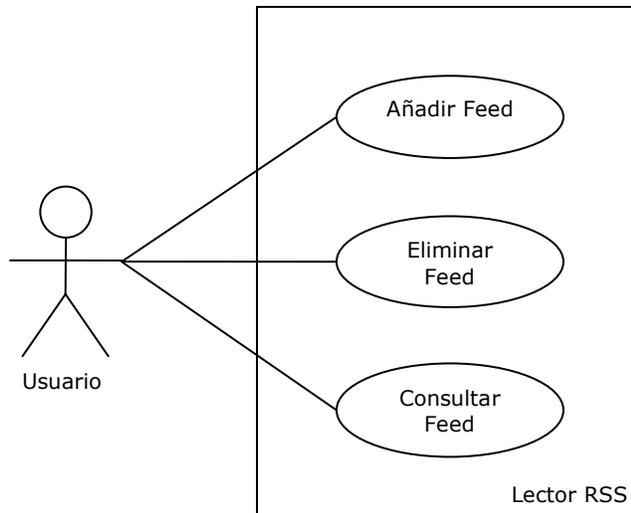


1.5.3. Diagrama de flujo para el proceso Analizar Feed



1.6. Diagramas de Casos de Uso

1.6.1. Diagrama de casos de uso para la gestión de feeds

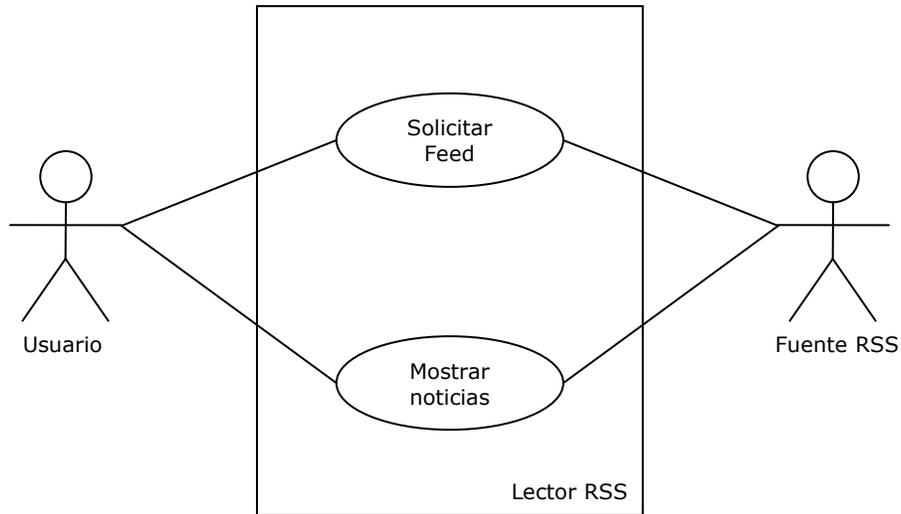


Caso de uso: Añadir Feed	
Actores	Usuario
Tipo	Primario
Descripción	El usuario puede asignar un nombre y una URL a un nuevo elemento Feed, que le permiten al Lector RSS acceder posteriormente realizando una conexión con la dirección URL.
Caso de uso anterior	Ninguno

Caso de uso: Eliminar Feed	
Actores	Usuario
Tipo	Primario
Descripción	El usuario elimina un feed creado anteriormente.
Caso de uso anterior	Añadir Feed

Caso de uso: Consultar Feed	
Actores	Usuario
Tipo	Primario
Descripción	El usuario consulta los datos de un feed creado anteriormente.
Caso de uso anterior	Añadir Feed

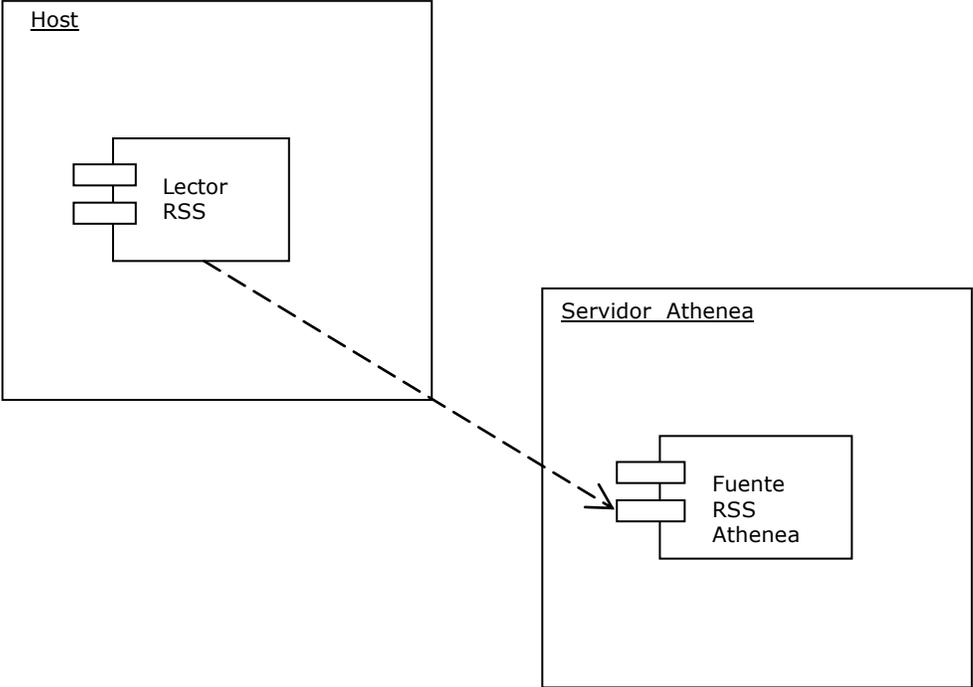
1.6.2. Diagrama de casos de uso para la conexión con la fuente RSS



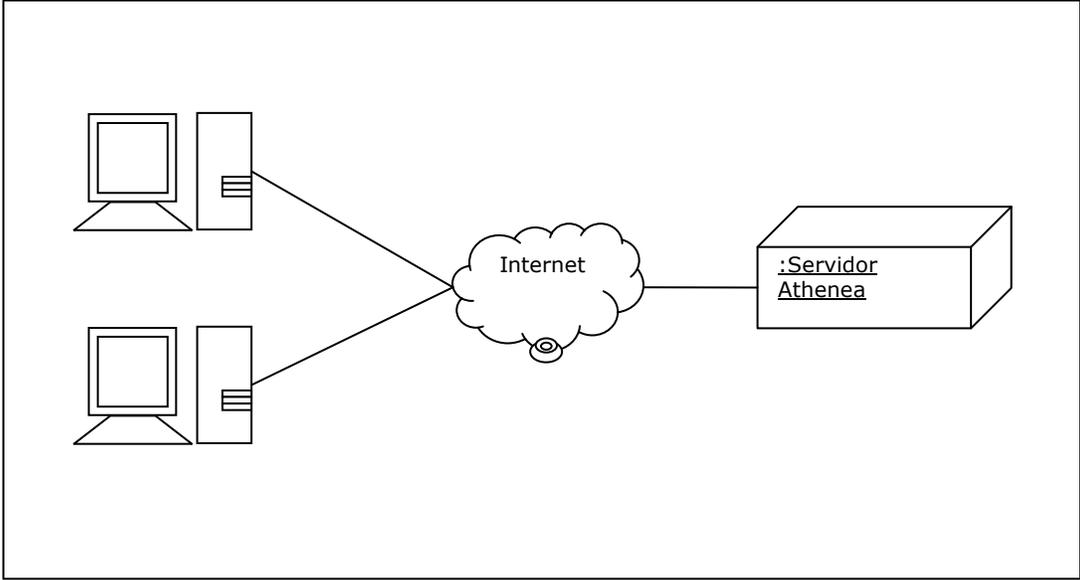
Caso de uso: Solicitar feed	
Actores	Usuario, Fuente RSS
Tipo	Primario
Descripción	El usuario solicita un feed a la fuente, enviando los datos de este, añadidos previamente.
Caso de uso anterior	Añadir Feed

Caso de uso: Mostrar noticias	
Actores	Usuario, Fuente RSS
Tipo	Primario
Descripción	La fuente RSS envía el feed en XML, para que la aplicación lo muestre al usuario de forma comprensible.
Caso de uso anterior	Solicitar Feed

1.7. Diagrama de componentes



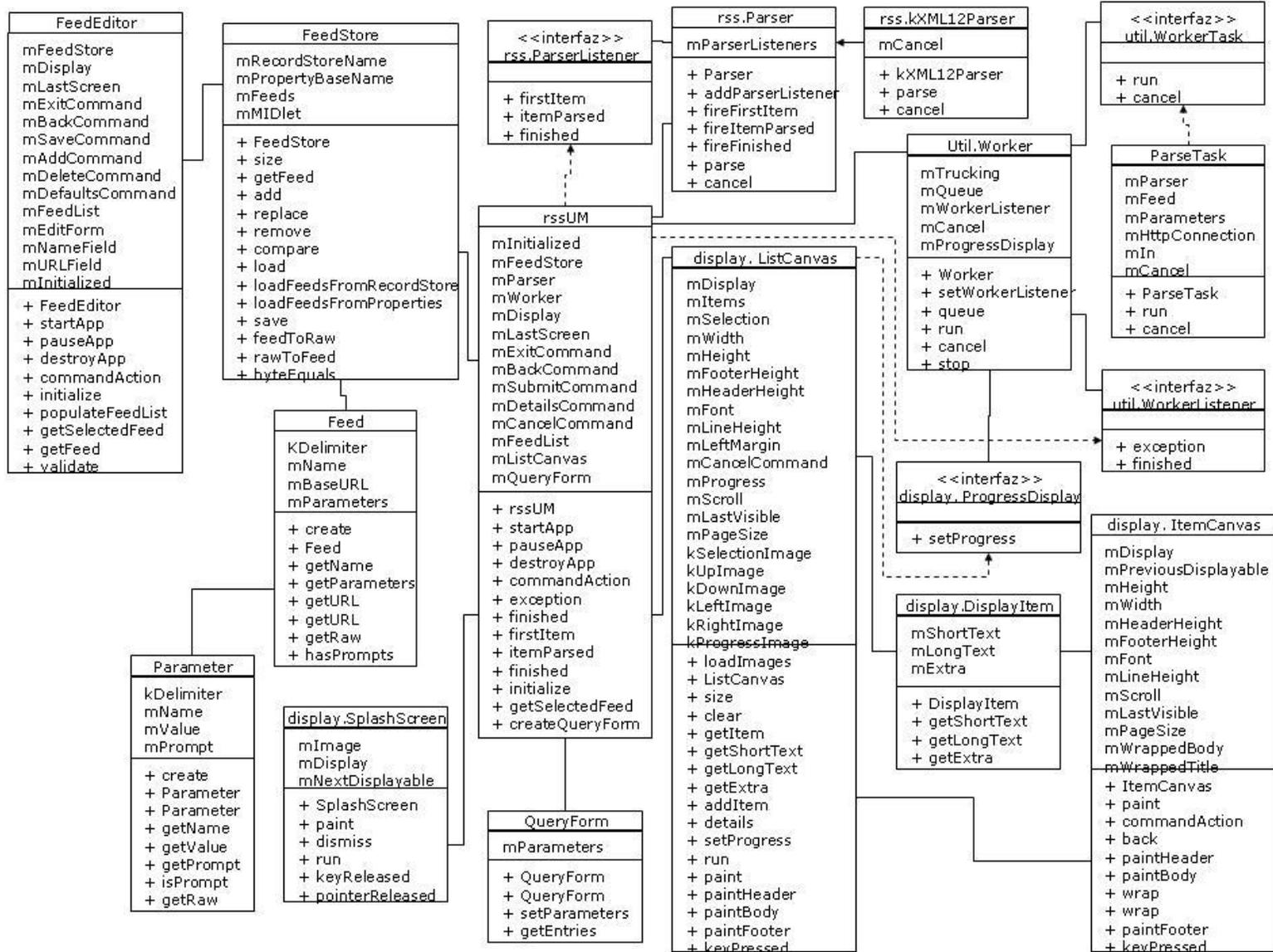
1.8. Diagrama de despliegue



2. Diseño Módulo Lector de Noticias RSSUM

2.1 Diagrama de Clases

Figura de la página siguiente



2.1.1 Diccionario de datos

Clase: Feed.java		
Descripción:	Representa un feed RSS o RDF	
Atributo	Descripción	Tipo dato
kDelimiter	Separa el nombre del feed con la descripción	Char
mName	Contiene el nombre del feed	String
mBaseURL	Contiene la dirección del feed	String
mParameters	Contiene los parámetros del feed	Vector
Métodos	Características	
create	Crea un objeto de tipo feed con los datos de la cadena dada.	
Feed	Constructor de la clase	
getName	Devuelve el nombre del feed	
getParameters	Devuelve los parámetros del feed	
getURL	Devuelve la url del feed si no requiere parámetros	
getURL	Devuelve la url del feed si requiere parámetros	
getRaw	Convierte el feed en un String para luego darle el formato estipulado por la aplicación	
hasPrompts	Devuelve "true" si alguno de los parámetros del feed necesita inserción de datos	

Clase: FeedEditor.java		
Descripción:	MIDlet que maneja la edicion de la lista de "feeds" de la aplicación	
Atributo	Descripción	Tipo dato
mFeedStore	Objeto del tipo FeedStore	FeedStore
mDisplay	Objeto del tipo Display	Display
mLastScreen	Funciona como un apuntador a la pantalla anterior	Displayable
mExitCommand	Comando para salir del MIDlet	Command
mBackCommand	Comando para ir hacia atrás	Command
mSaveCommand	Comando para guardar cambios	Command
mDefaultsCommand	Comando para dejar la lista de feeds en su forma original	Command
mDeleteCommand	Comando para eliminar feeds	Command
mAddCommand	Comando para agregar feeds	Command
mFeedList	Objeto del tipo List	List
mEditForm	Objeto del tipo Form	Form
mNameField	Campo de texto para el nombre del feed	TextField
mURLField	Campo de texto para la dirección del feed	TextField
mEditFeed	Objeto del tipo Feed	Feed
mInitialized	Permite definir si se ha editado la lista de feeds	boolean
Métodos	Características	
FeedEditor	Constructor de la clase	
startApp	Asigna la pantalla y valida si se ha inicializado	
pauseApp	Asigna al atributo mLastScreen la pantalla actual	
destroyApp	Destructor	
commandAction	Gestiona los datos del commandListener	

initialize	Inicializa todos los elementos del editor de feeds
populateFeedList	Publica la lista de feeds
getSelectedFeed	Devuelve el feed seleccionado
getFeed	Devuelve el feed identificado por el parámetro name
validate	Valida que el nombre y la url estén escritos correctamente

Clase: FeedStore.java		
Descripción:	Representa una lista de feeds que son almacenados en un objeto de tipo recordStore	
Atributo	Descripción	Tipo dato
mRecordStoreName	El nombre del record store del cual el feed debe ser cargado	String
mPropertyBaseName	Una base que será usada para cargar los feeds por defecto si el record store no está presente	String
mFeeds	Vector en el que se cargan los feeds	Vector
mMIDlet	Representa el MIDlet en el cual se encuentran los parámetros por defecto necesarios para el inicio de la aplicación.	MIDlet
Métodos	Características	
FeedStore	Constructor de la clase	
size	Devuelve el número de Feeds que hay en el FeedStore	
getFeed	Devuelve el feed correspondiente al índice dado	
add	Agrega un nuevo feed	
replace	Reemplaza un feed	
remove	Elimina un feed	
compare	Compara dos feeds para saber si han sido editados	
load	Carga una lista de feeds	
loadFeedsFromRecordStore	Carga la lista de feeds de un record store	
loadFeedsFromProperties	Carga la lista de feeds de las propiedades del MIDlet	
save	Guarda los feeds en el record store	
feedToRaw	Convierte un feed a una cadena de bytes	
rawToFeed	Convierte una cadena de bytes a un objeto feed	
byteEquals	Compara dos cadenas de bytes que representan feeds	

Clase: Parameter.java		
Descripción:	Representa un parámetro del feed	
Atributo	Descripción	Tipo dato
kDelimiter	Separa los tres tipos de parámetros	Char
mName	El nombre del feed	String
mValue	La URL del feed	String
mPrompt	Contiene un parámetro que el usuario ingrese	String
Métodos	Características	
create	Crea un objeto Parámetro con una cadena dada	
Parameter	Crea un parámetro sin intervención del usuario	
Parameter	Crea un parámetro con intervención del usuario	

getName	Devuelve el nombre del feed
getValue	Devuelve la url del feed
getPrompt	Devuelve el parámetro ingresado por el usuario
isPrompt	Devuelve verdadero si el parámetro tiene datos ingresados por el usuario
getRaw	Devuelve una cadena con el parámetro

Clase: ParseTask.java		
Descripción:	Puede leer un feed RSS o RDF	
Atributo	Descripción	Tipo dato
mParser	Objeto del tipo Parser	Parser
mFeed	Objeto del tipo Feed	Feed
mParameters	Contiene los parámetros	Vector
mHttpConnection	Encargado de la conexión	HttpConnection
mIn	Contiene el flujo de entrada de la conexión http	InputStream
mCancel	Identifica si se ha cancelado un análisis de feed	boolean
Métodos	Características	
ParseTask	Constructor de la clase	
run	Se conecta a un feed y lo analiza	
cancel	Cancela el análisis actual	

Clase: QueryForm.java		
Descripción:	Esta clase es usada para pedir los parámetros al usuario	
Atributo	Descripción	Tipo dato
mParameters	Una lista de parámetros	Vector
Métodos	Características	
QueryForm	Crea un formulario con una lista de parámetros vacía	
QueryForm	Crea un formulario con un título y una lista de parámetros	
setParameters	Configura la lista de parámetros para ser mostrada en el formulario	
getEntries	Devuelve una lista de parámetros representados en el formulario	

Clase: rssUM.java		
Descripción:	Es el MIDlet que controla la aplicación rssUM	
Atributo	Descripción	Tipo dato
mInitialized	Permite definir si se ha editado la lista de feeds	Boolean
mFeedStore	Objeto del tipo FeedStore	FeedStore
mParser	Objeto del tipo Parser	Parser
mWorker	Objeto del tipo Worker	Worker
mDisplay	Objeto del tipo Display	Display
mLastScreen	Funciona como apuntador a una pantalla anterior	Displayable
mExitCommand	Comando para salir de la aplicación	Command
mBackCommand	Comando para volver a la lista de feeds	Command
mSubmitCommand	Comando utilizado cuando se requiere el ingreso de parámetros por parte del usuario	Command

mDetailsCommand	Comando para ver los detalles de un feed seleccionado	Command
mCancelCommand	Comando que cancela una tarea ejecutada por el objeto de tipo Worker	Command
mFeedList	Lista de Feeds que se van a mostrar	List
mListCanvas	Objeto del tipo ListCanvas	ListCanvas
mQueryForm	Objeto del tipo QueryForm	QueryForm
Métodos	Características	
rssUM	Constructor de la clase	
startApp	Inicia la aplicación	
pauseApp	Guarda la ultima ventana visible, que será restaurada en startApp	
destroyApp	Destructor	
commandAction	Gestiona la información del comandListener	
exception	Genera una excepción cuando se intenta leer un feed.	
finished	Este método se llama cuando se termina de ejecutar el método exception	
firstItem	Este método se llama cuando el primer item es analizado	
itemParsed	Este método se llama cada vez que un item es analizado	
finished	Este método pone una pantalla que indica que no se encontraron ítems	
initialize	Inicia todos los elementos de la aplicación rssUM	
getSelectedFeed	Devuelve un feed seleccionado	
createQueryForm	Crea un formulario para ingresar parámetros	

Clase: display.DisplayItem.java		
Descripción:	Define tres atributos: texto corto, texto largo y texto extra y los métodos que los reciben	
Atributo	Descripción	Tipo dato
mShortText	Contiene el titular de cada noticia por feed	String
mLongText	Contiene el contenido de cada noticia por feed	String
mExtra	Contiene el texto extra de cada noticia por feed	String
Métodos	Características	
DisplayItem	Método en el que se asignan los tres tipos de texto a sus respectivos atributos	
getShortText	Devuelve el texto corto	
getLongText	Devuelve el texto largo	
getExtra	Devuelve el texto extra	

Clase: display.ItemCanvas.java		
Descripción:	Muestra el texto corto y el texto largo de un item del feed	
Atributo	Descripción	Tipo dato
mDisplay	Objeto del tipo Display	Display
mPreviousDisplayable		Displayable
mWidth	Define el ancho del área disponible para la aplicación	Int

mHeight	Define la altura del área disponible para la aplicación	int
mHeaderHeight	Define la altura del encabezado	Int
mFooterHeight	Define la altura del pie de página	Int
mFont	Objeto del tipo Font	Font
mLineHeight	Contiene la altura de la fuente	Int
mScroll	Define si se rebasa el tamaño de la pantalla	Int
mLastVisible	Define el último item visible	Int
mPageSize	Define la cantidad de pantallas en tamaño que deben ocuparse para mostrar todo el texto	Int
mWrappedTitle	Contiene el titular del feed	Vector
mWrappedBody	Texto completo del contenido de cada feed	Vector
Métodos	Características	
ItemCanvas	Constructor de la clase	
Saint	Gestiona la interfaz de lectura	
commandAction	Gestiona la información del comandListener	
Back	Devuelve la aplicación a una pantalla previa	
paintHeader	Pinta la cabecera o titular de la noticia	
paintBody	Pinta el cuerpo de la noticia seleccionada	
wrap	Solicita el titular y el cuerpo de la noticia o elemento del feed	
wrap	Da formato al titular y al cuerpo de la noticia o elemento del feed	
paintFooter	Pinta el pie de pagina de la noticia	
keyPressed	Es un keyListener	

Clase: display.ListCanvas.java		
Descripción:	Representa una pantalla de ítems	
Atributo	Descripción	Tipo dato
mDisplay	Objeto del tipo Display	Display
mItems	Lista de ítems	Vector
mSelection	Apunta a un ítem seleccionado	Int
mWidth	Define el ancho del área disponible para la aplicación	Int
mHeight	Define el alto del área disponible para la aplicación	Int
mHeaderHeight	Define la altura del encabezado	Int
mFooterHeight	Define la altura del pie de página	Int
mFont	Objeto del tipo Font	Font
mLineHeight	Contiene la altura de la fuente	Int
mLeftMargin	Contiene el ancho de la imagen de selección de ítems	Int
mCancelCommand	Comando cancelar mostrado al mismo tiempo que la barra de progreso	Command
mProgress	Controla el tiempo en que la barra de progreso es visible	Int
mScroll	Define si se rebasa el tamaño de la pantalla	Int

mLastVisible	Define el último item visible	Int
mPageSize	Define la cantidad de pantallas en tamaño que deben ocuparse para mostrar todo el texto	Int
kSelectionImage	Identifica un mensaje seleccionado	Image
kUpImage	Indica que se puede ir hacia arriba	Image
kDownImage	Indica que se puede ir hacia abajo	Image
kLeftImage	Indica que se puede ir a la izquierda	Image
kRightImage	Indica que se puede ir a la derecha	Image
kProgressImage	Contiene la barra de progreso	Image
Métodos	Características	
loadImages	Carga las imágenes utilizadas	
listCanvas	Constructor de la clase	
size	Devuelve el numero de ítems de la lista	
clear	Remueve todos los ítems de la lista	
getItem	Devuelve el item seleccionado	
getShortText	Devuelve el texto corto o titular de un item seleccionado	
getLongText	Devuelve el texto largo o contenido de un item seleccionado	
getExtra	Devuelve el texto extra de un item seleccionado	
addItem	Añade un nuevo item a la lista	
Details	Muestra la pantalla de detalles para el item seleccionado	
setProgress	Define si se utiliza la barra de progreso y el comando cancelar	
run	Ejecuta un ciclo para animar la barra de progreso	
paint	Pinta los ítems de la lista	
paintHeader	Pinta la cabecera de la noticia	
paintBody	Pinta el cuerpo de la noticia	
paintFooter	Pinta el pie de pagina la noticia	
keyPressed	Es un keyListener	

Clase: <<interfaz>> display.ProgressDisplay.java		
Descripción:	Interfaz para un objeto que muestra un indicador visual de trabajo	
Atributo	Descripción	Tipo dato
Métodos	Características	
setProgress	Se usa para activar o desactivar el indicador de progreso	

Clase: display.SplashScreen.java		
Descripción:	Crea una pantalla temporal	
Atributo	Descripción	Tipo dato
mImage	Contiene la imagen que va a ser mostrada en la pantalla temporal	Image
mDisplay	Objeto del tipo Display	Display
mNextDisplayable	La pantalla que se mostrará al terminar esta pantalla temporal	Displayable
Métodos	Características	
SplashScreen	Constructor de la clase	
paint	Pinta la imagen centrada en la pantalla	

dismiss	Descarta la pantalla temporal
run	Define la duración de la pantalla temporal
keyReleased	Llama a dismiss cuando una tecla es soltada
pointerReleased	Llama a dismiss cuando un puntero es soltado

Clase: rss.Parser.java		
Descripción:	Superclase abstracta para objetos que analizan archivos RSS o RDF	
Atributo	Descripción	Tipo dato
mParserListeners	Lista de objetos listener	Vector
Métodos	Características	
Parser	Constructor de la clase	
addParserListener	Añade un nuevo objeto Listener	
fireFirstItem	Las subclases llaman a este método para acceder al método firstItem	
fireItemParsed	Las subclases llaman a este método para acceder al método itemParsed	
fireFinished	Las subclases llaman a este método para acceder al método finished	
parse	Las subclases deben implementar este método para analizar un documento de un flujo de entrada dado	
cancel	Las subclases deben implementar este método para cancelar el análisis	

Clase: <<interfaz>> rss.ParserListener.java		
Descripción:	Interfaz implementada para objetos que esperan eventos de una subclase Parser	
Atributo	Descripción	Tipo dato
Métodos	Características	
firstItem	Método llamado cuando el primer item es analizado	
itemParsed	Método llamado cada vez que un item es analizado	
finished	Método llamado cuando el análisis ha terminado	

Clase: rss.kXML12Parser.java		
Descripción:	Analiza documentos RDF y RSS 0.91	
Atributo	Descripción	Tipo dato
mCancel	Define si se cancela la lectura de un documento	Boolean
Métodos	Características	
kXML12Parser	Constructor de la clase	
parse	Realiza el análisis de un documento RDF o RSS	
cancel	Cancela el análisis del documento	

Clase: util.Worker.java		
Descripción:	Clase que puede correr tareas dentro de un hilo dedicado y separado	
Atributo	Descripción	Tipo dato

mTrucking	Indica cuando hay un proceso en ejecución	boolean
mQueue	Lista de tareas	Vector
mWorkerListener	Objeto del tipo WorkerListener	WorkerListener
mCancel	Indica si se debe cancelar una tarea	boolean
mProgressDisplay	Muestra un indicador mientras una tarea se esta ejecutando	ProgressDisplay
Métodos	Características	
Worker	Constructor de la clase	
setWorkerListener	Asigna un objeto listener	
queue	Adiciona tareas a la pila	
run	Método llamado por el constructor para ejecutar tareas	
cancel	Cancela la tarea que se este ejecutando	
stop	Detiene todas las tareas	

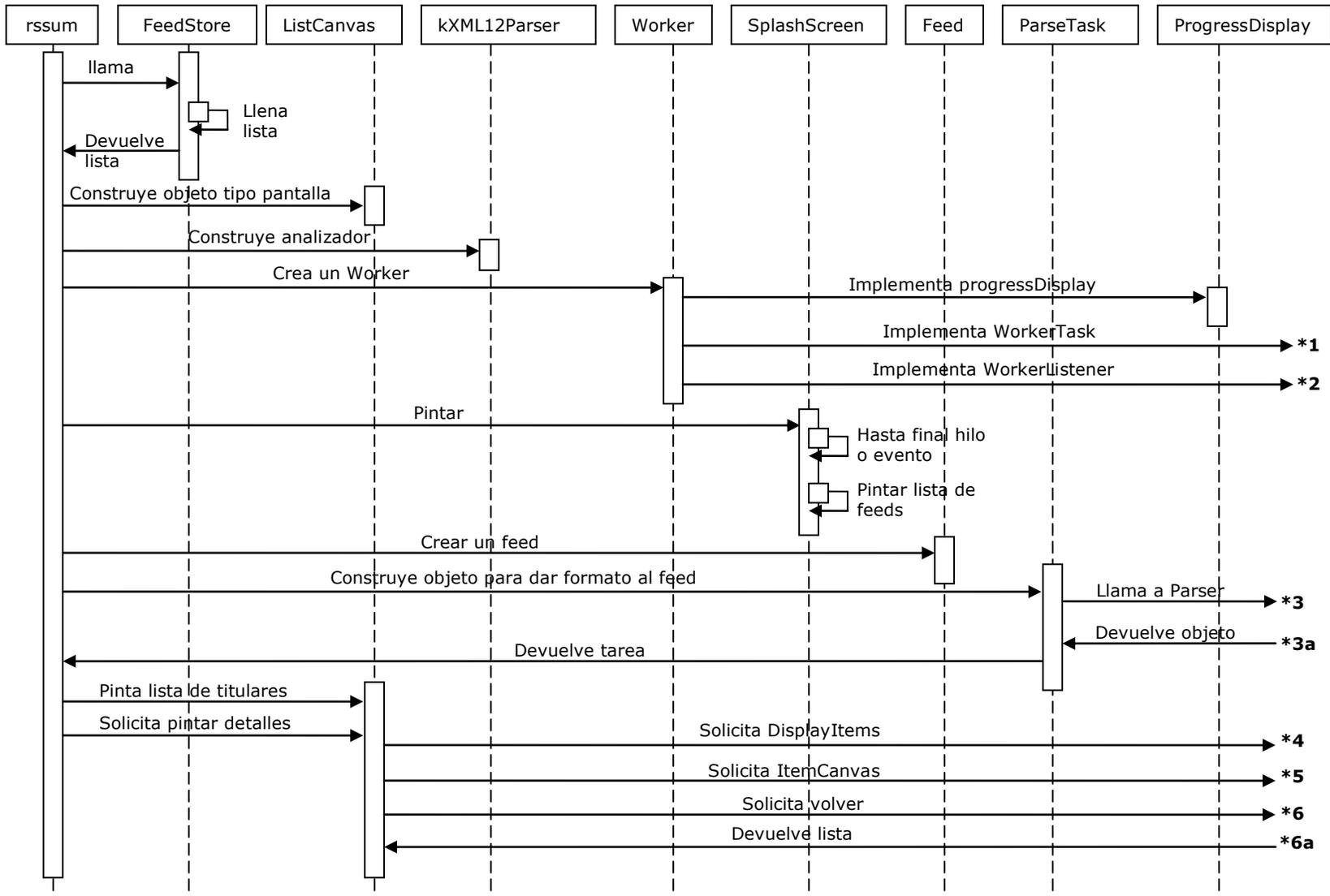
Clase: <<interfaz>> util.WorkerListener.java		
Descripción:	Interfaz para objetos que reciben notificaciones de objetos Worker y sus tareas	
Atributo	Descripción	Tipo dato
Métodos	Características	
exception	Método llamado cuando ocurre una excepción en el método run del WorkerTask	
finished	Método llamado cuando termina el método run del WorkerTask	

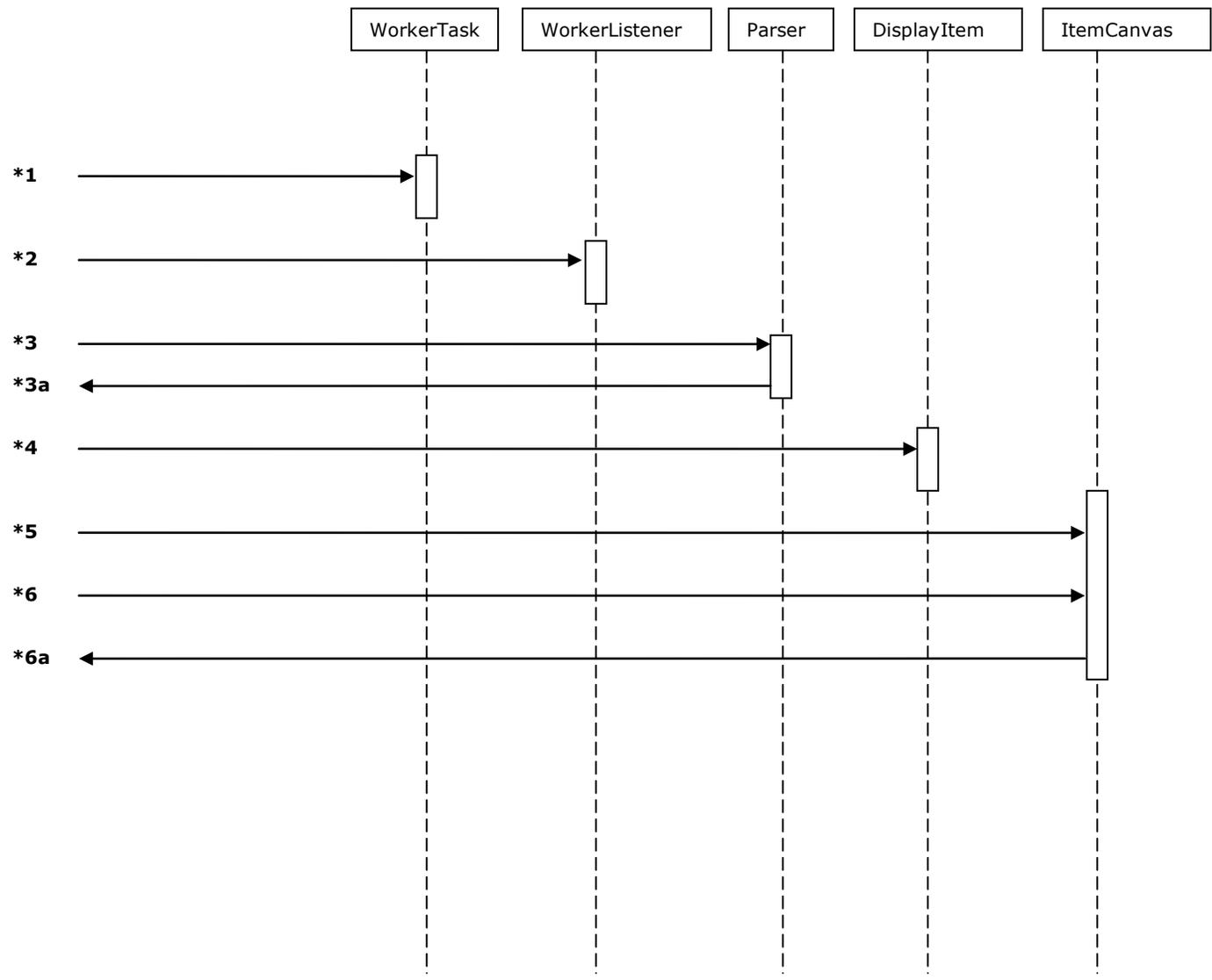
Clase: <<interfaz>> util.WorkerTask.java		
Descripción:	Interfaz implementada por las tareas que pueden ser ejecutadas	
Atributo	Descripción	Tipo dato
Métodos	Características	
run	Método llamado dentro de un hilo de Worker	
cancel	Puede ser llamado en cualquier momento. Es el responsable de que las tareas terminen satisfactoriamente	

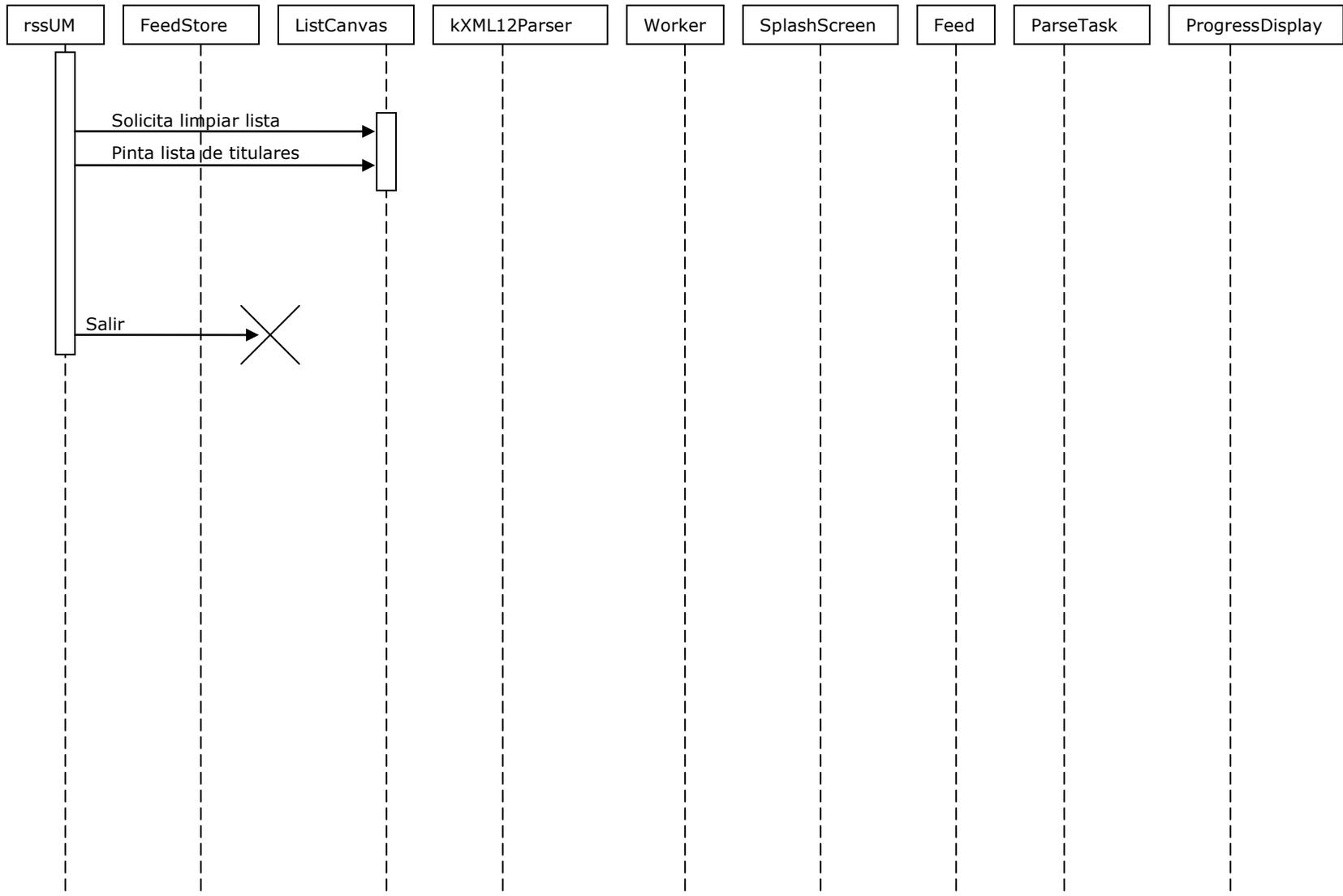
2.2. Diagrama de secuencia

2.2.1. Diagrama de secuencia para la aplicación rssUM

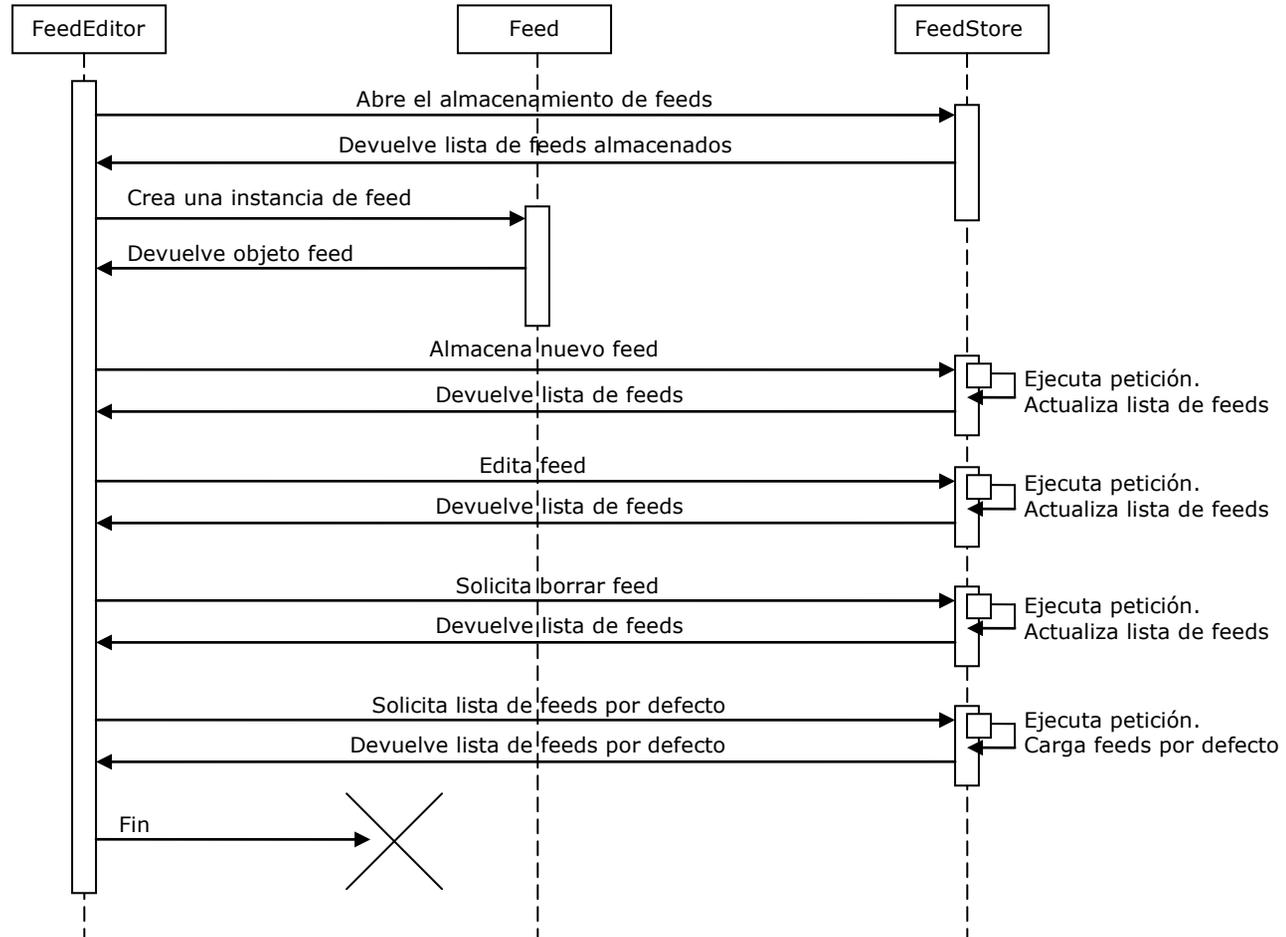
Página siguiente





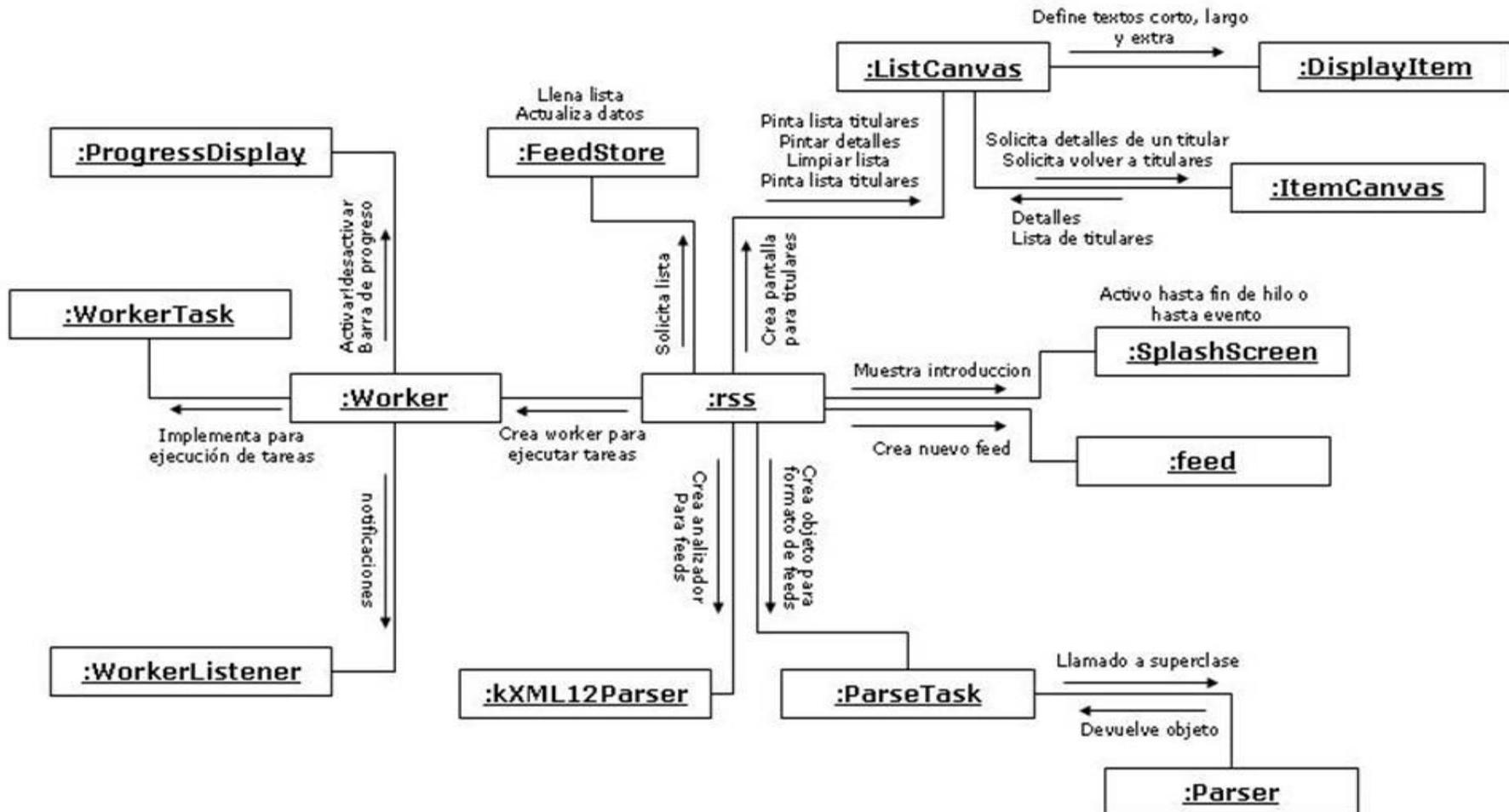


2.2.2 Diagrama de secuencia para la aplicación FeedEditor (Editor de Feeds)

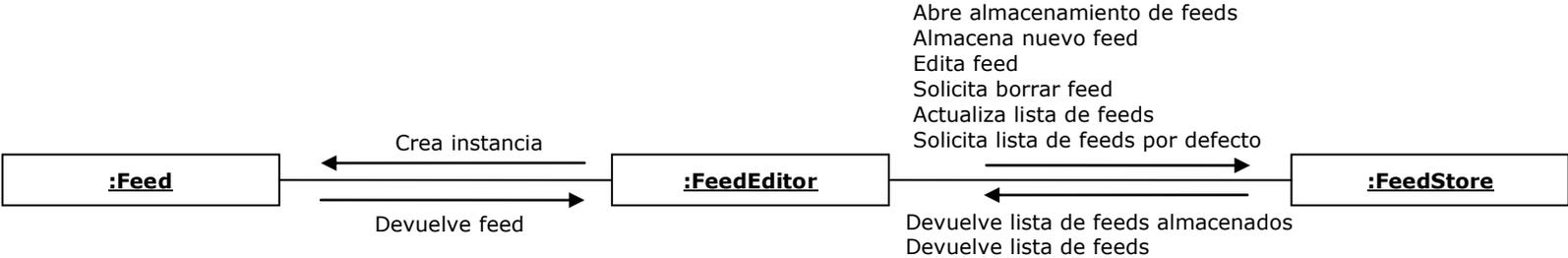


2.3 Diagrama de colaboración

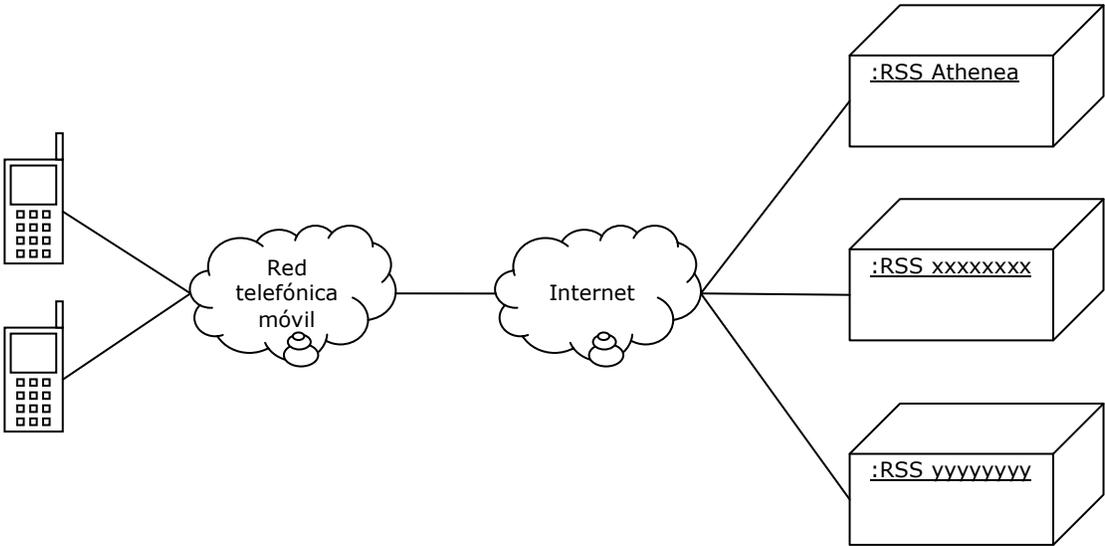
2.3.1 Diagrama de colaboración para la aplicación RSSUM



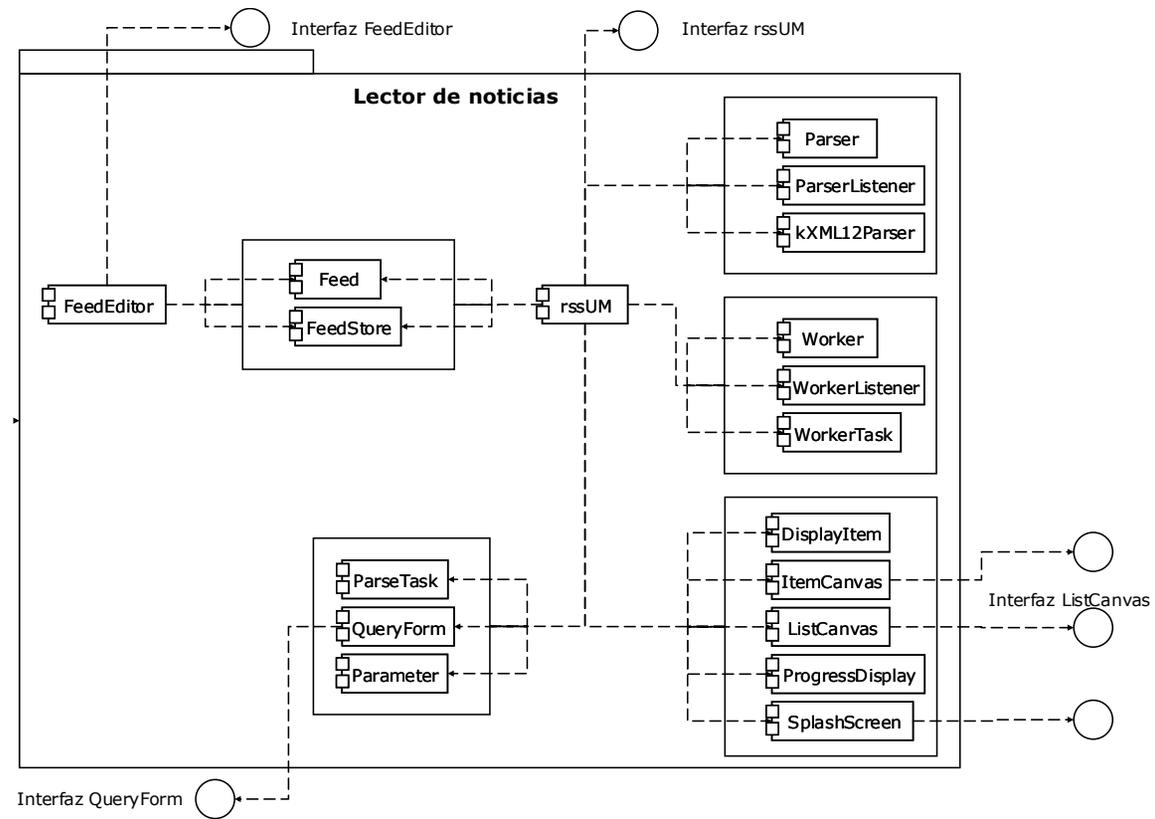
2.3.2 Diagrama de colaboración para la aplicación FeedEditor



2.4 Diagrama de despliegue



2.5 Diagrama de componentes



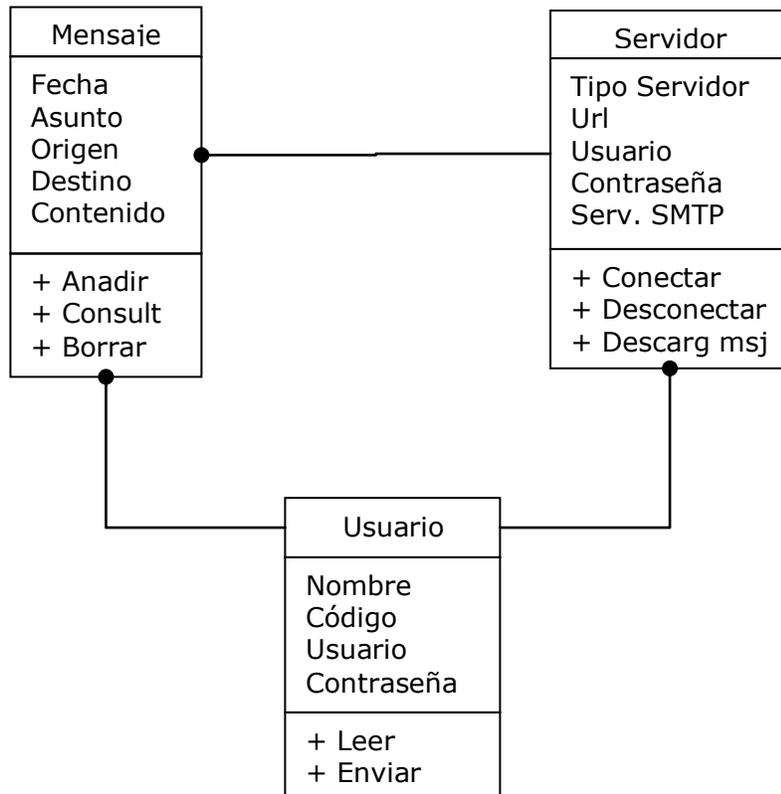
ANEXO B

ANÁLISIS Y DISEÑO DEL MÓDULO CORREOUM

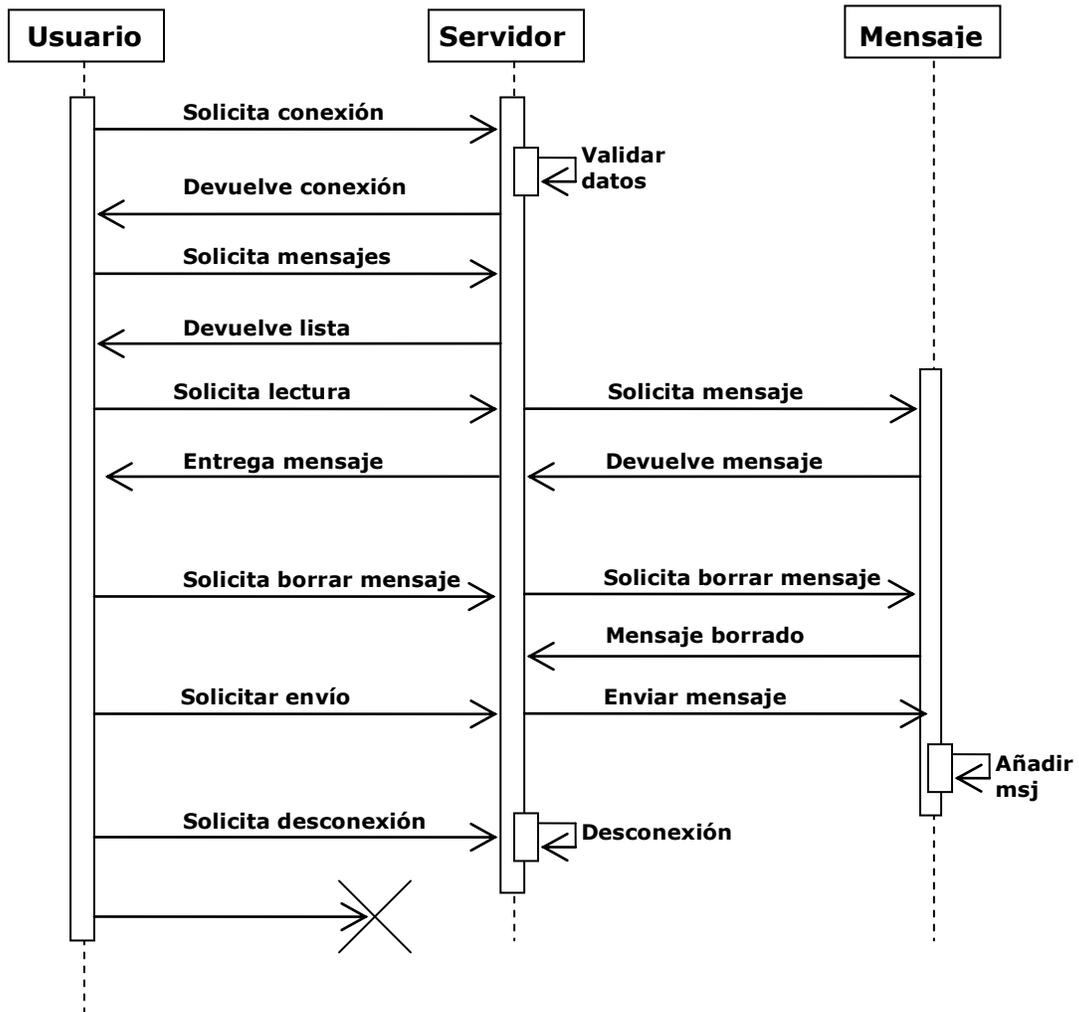
Descripción: Este módulo es un cliente de correo ligado al servidor de correo de athenea; ejecuta las funciones de envío, reenvío y respuesta, además de consultar y eliminar los correos.

1. Análisis Módulo de Correo CorreoUM

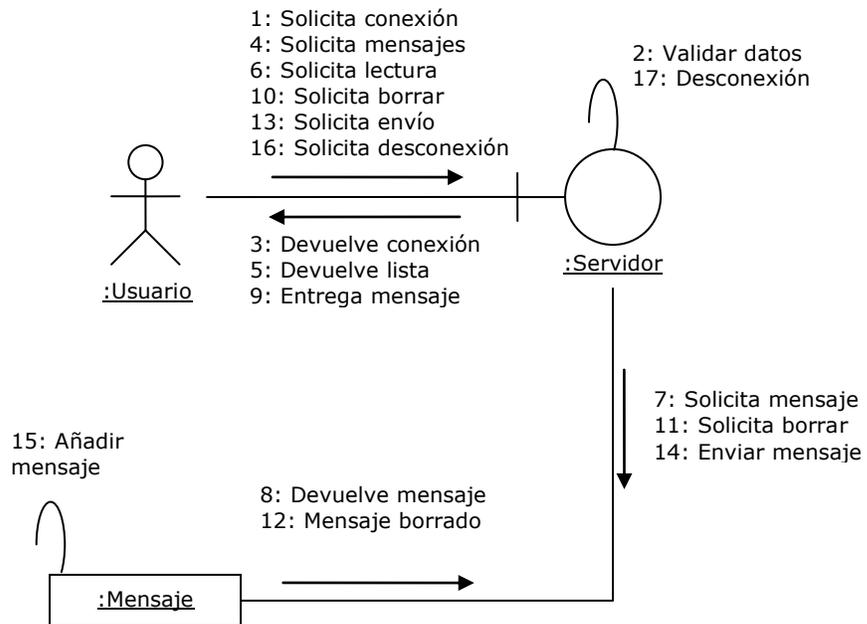
1.1 Diagrama de Clases



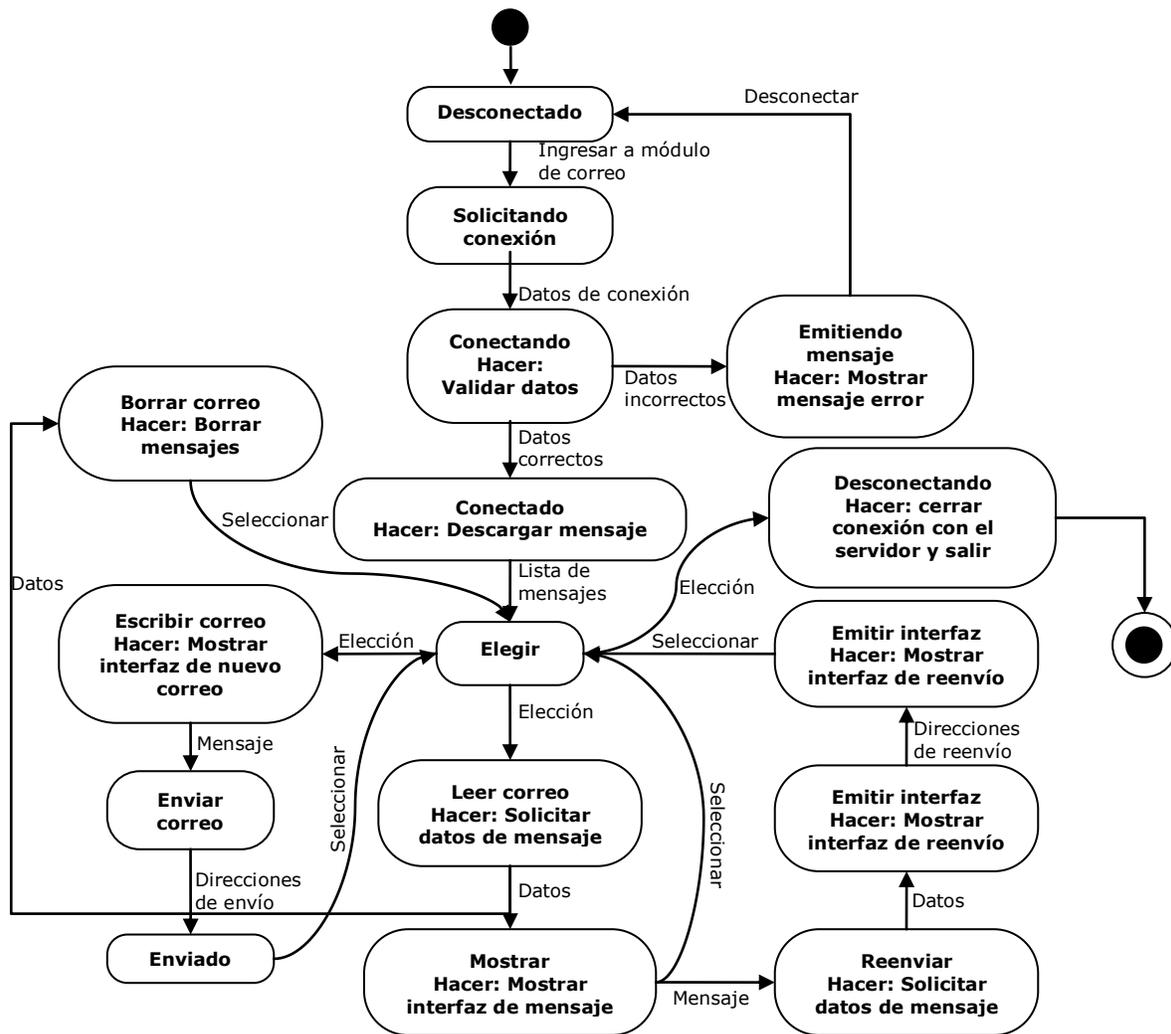
1.2 Diagrama de Sucesos



1.3 Diagrama de Colaboración

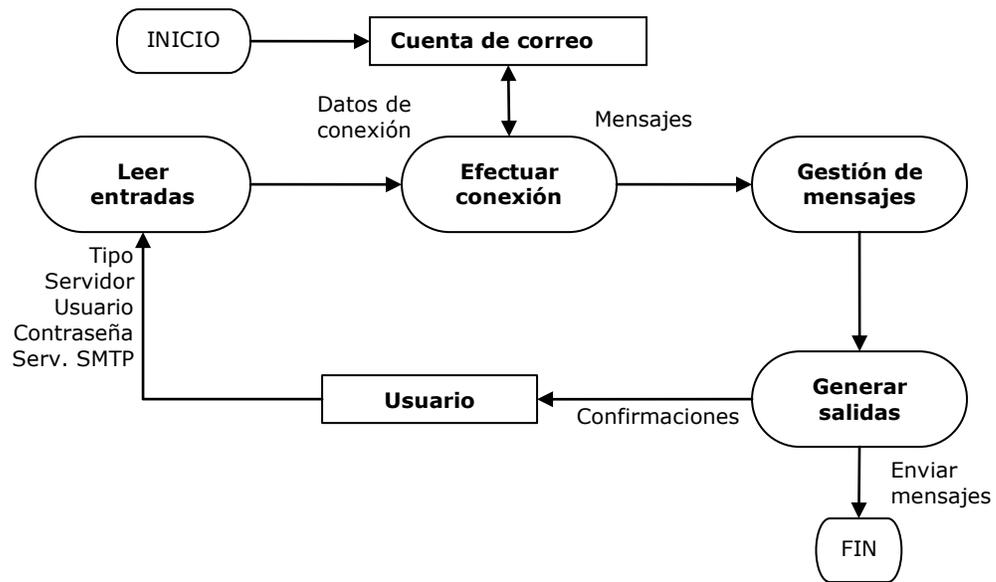


1.4 Diagrama de Estados

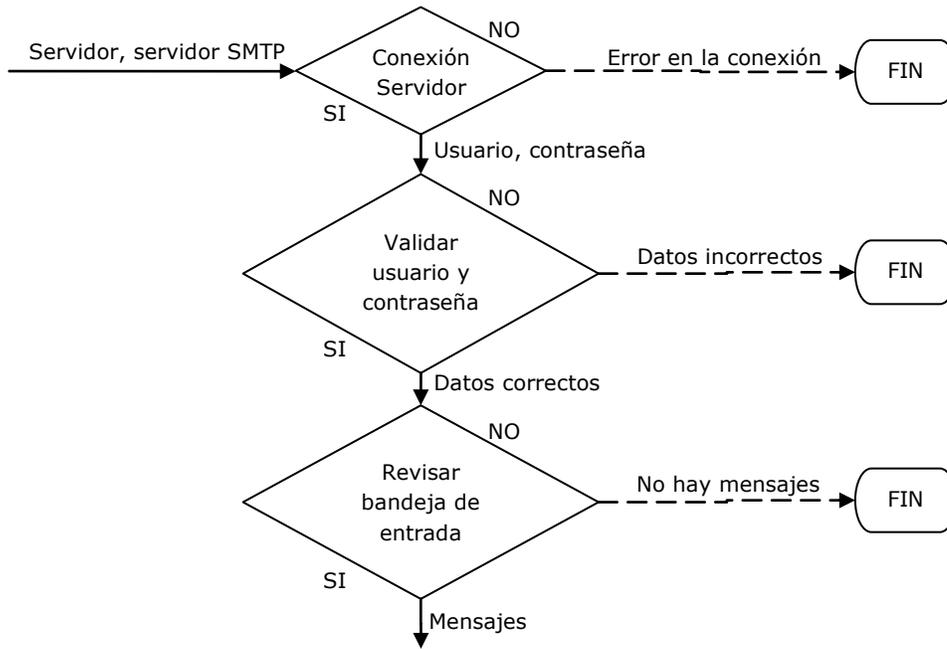


1.5 Diagrama de Flujo

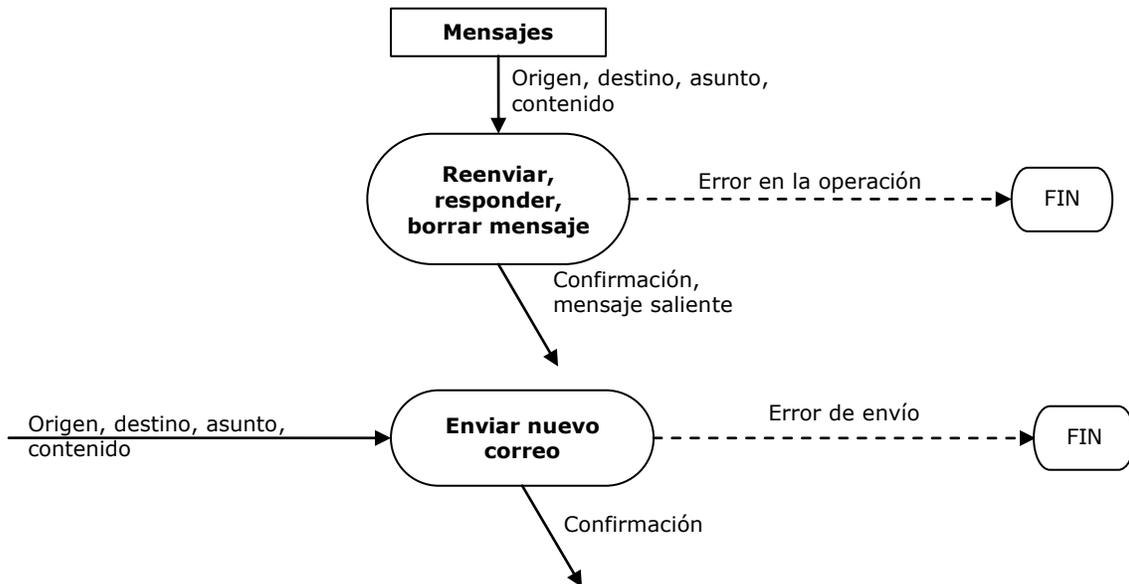
1.5.1 Diagrama de flujo general



1.5.2 Diagrama de flujo para el proceso efectuar conexión

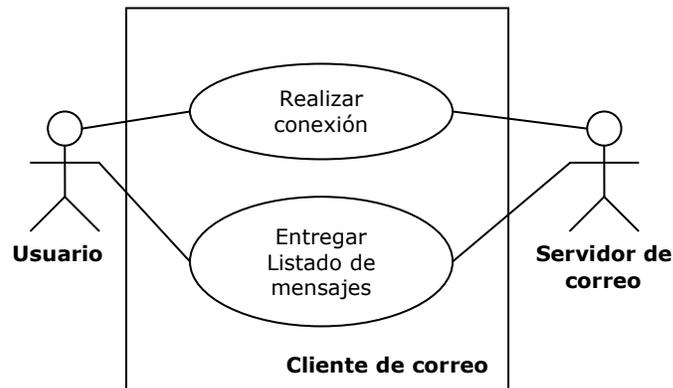


1.5.3 Diagrama de flujo para el proceso Gestión de mensajes



1.6 Diagrama de Casos de Uso

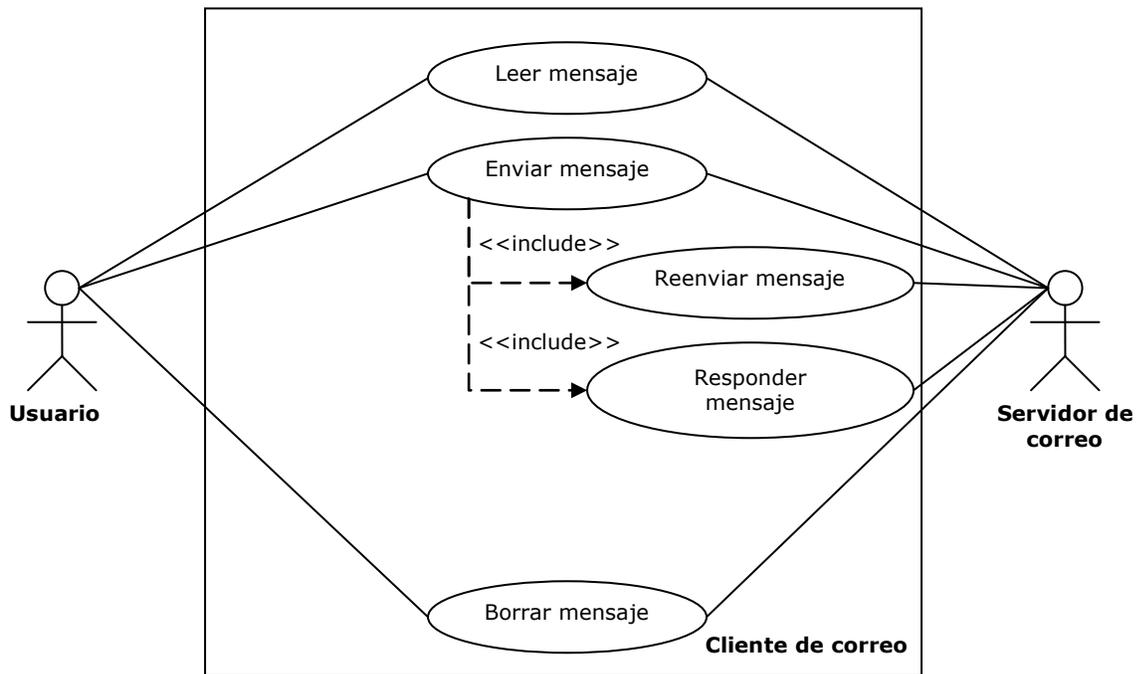
1.6.1 Diagrama de Casos de Uso para la conexión con el servidor de correo



Caso de uso: Realizar conexión	
Actores	Usuario, Servidor de correo
Tipo	Primario
Descripción	El usuario envía petición de conexión con sus datos. El servidor le devuelve la conexión al usuario.
Caso de uso anterior	Ninguno

Caso de uso: Listado de mensajes	
Actores	Usuario, Servidor de correo
Tipo	Primario
Descripción	El servidor de correo envía el listado de mensajes que el usuario tenga en su cuenta de correo.
Caso de uso anterior	Ninguno

1.6.2 Diagrama de Casos de Uso para la gestión de mensajes



Caso de uso: Leer mensaje	
Actores	Usuario, Servidor de correo
Tipo	Primario
Descripción	El usuario selecciona un mensaje de la lista y el servidor de correo devuelve el cuerpo del mensaje.
Caso de uso anterior	Ninguno

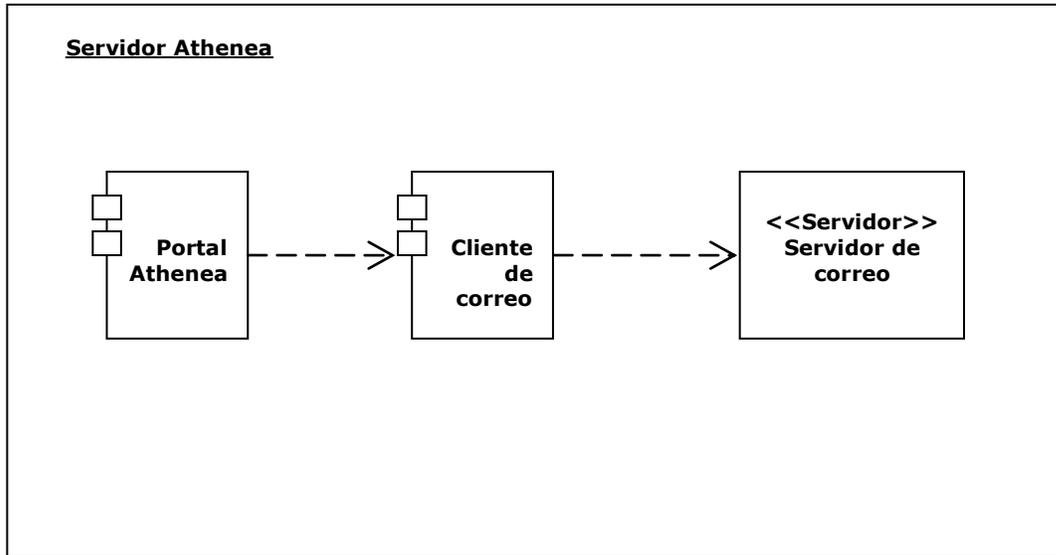
Caso de uso: Enviar mensaje	
Actores	Usuario, Servidor de correo
Tipo	Primario
Descripción	El usuario crea un nuevo mensaje para enviar a un destinatario cualquiera. El servidor envía el mensaje.
Caso de uso anterior	Ninguno

Caso de uso: Reenviar mensaje	
Actores	Usuario, Servidor de correo
Tipo	Primario
Descripción	El usuario envía a uno o varios destinos un correo que tenga en su lista. El servidor envía el mensaje a todos los destinos.
Caso de uso anterior	Enviar mensaje

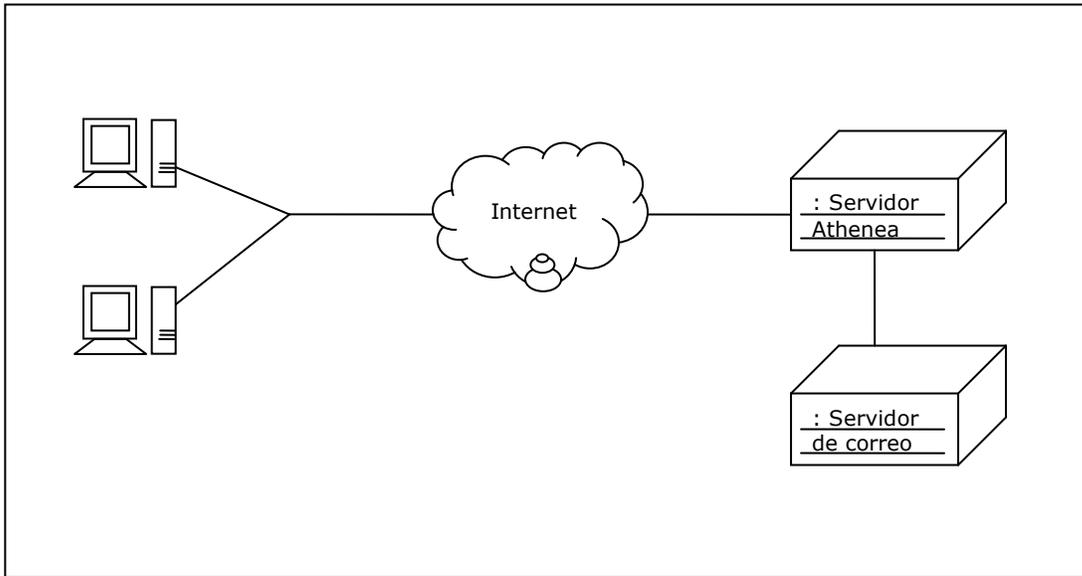
Caso de uso: Responder mensaje	
Actores	Usuario, Servidor de correo
Tipo	Primario
Descripción	El usuario responde a un destino basado en un mensaje de la lista. El servidor envía el correo al destino especificado en la lista de mensajes.
Caso de uso anterior	Enviar mensaje

Caso de uso: Borrar mensaje	
Actores	Usuario, Servidor de correo
Tipo	Primario
Descripción	El usuario selecciona un mensaje de la lista, el cual será borrado por el servidor de correo.
Caso de uso anterior	Ninguno

1.7 Diagrama de Componentes

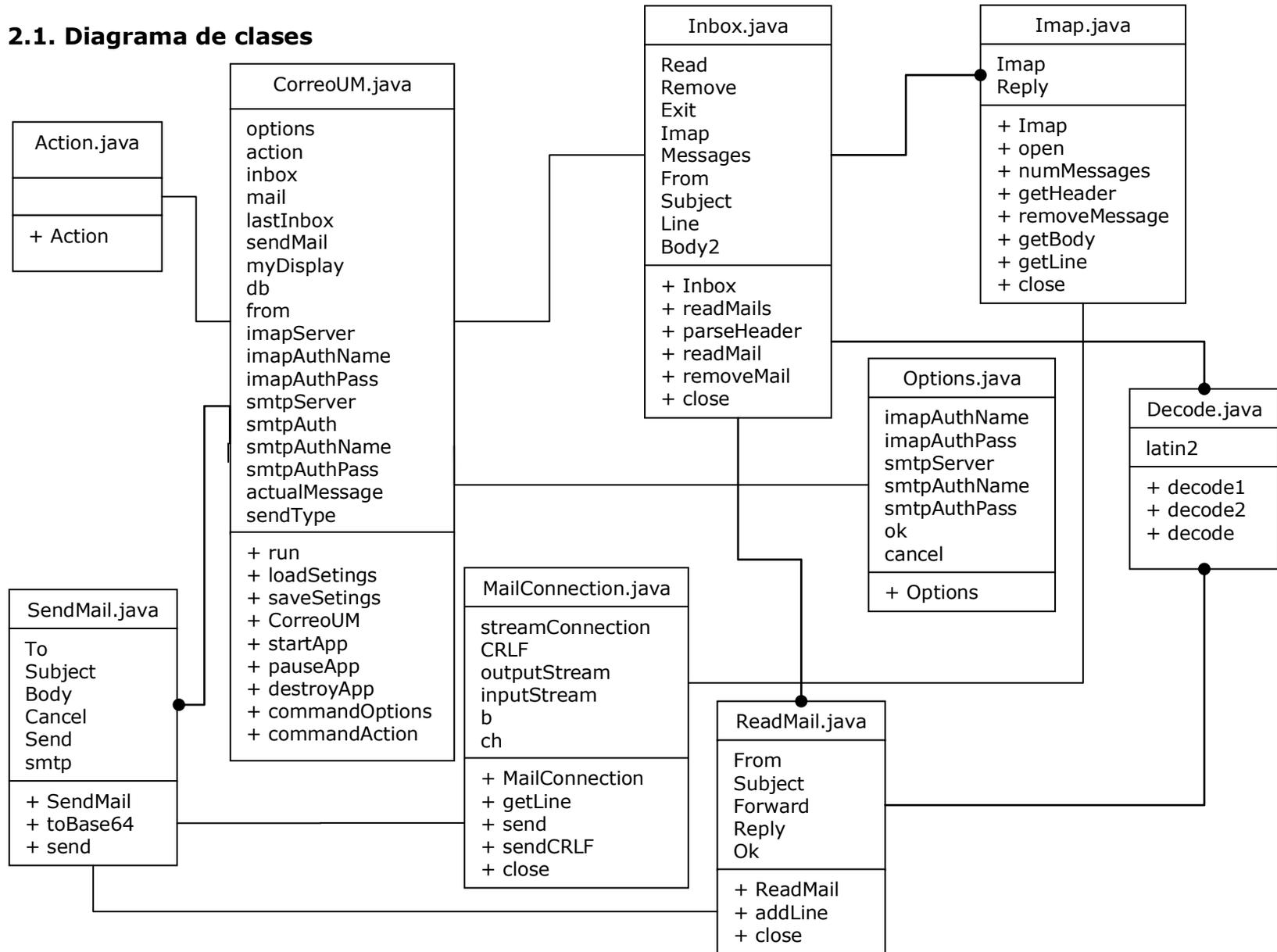


1.8 Diagrama de Despliegue



2. DISEÑO MODULO CORREO

2.1. Diagrama de clases



2.1.1 Diccionario de datos

Clase: Decode.java		
Descripción	Da formato al texto	
Atributo	Descripción	Tipo dato
Latin2	Atributo estático que contiene los símbolos de la codificación "latin"	Char
Métodos	Características	
decode1	Selecciona el texto para formatear	
decode2	Formatea el texto seleccionado por decode1	
decode	Ordena el texto formateado	

Clase: Action.java		
Descripción	Genera el menú principal del cliente de correo	
Atributo	Descripción	Tipo dato
Métodos	Características	
Action	Contiene las opciones del menú y les adiciona a estas el CommandListener	

Clase: ReadMail.java		
Descripción	Genera la interfaz para leer un correo	
Atributo	Descripción	Tipo dato
From	Etiqueta para recibir el origen del correo	String
Subject	Etiqueta para recibir el asunto del correo	String
Forward	Botón para reenviar un correo	Command
Reply	Botón para responder un correo	Command
Ok	Botón para volver a la bandeja de entrada	Command
Métodos	Características	
ReadMail	Constructor de la clase	
addLine	Adiciona una línea a la sección del cuerpo de la interfaz readMail	
close	Borra la interfaz	

Clase: SendMail.java		
Descripción	Se ocupa de todo el proceso de envío de correos	
Atributo	Descripción	Tipo dato
to	Campo para ubicar el destino del correo	TextField
subject	Campo para ubicar el asunto el correo	TextField
body	Campo para ubicar el cuerpo del correo	TextField
cancel	Botón para cancelar la acción de envío	Command
send	Botón para aceptar la acción de envío	Command
smtp	Crea un objeto del tipo MailConnection para realizar la conexión al protocolo smtp	MailConnection
Métodos	Características	
sendMail	Constructor de la clase, crea la interfaz de envío	
toBase64	Codifica a 64 bits los datos del usuario para cuestiones de seguridad	
send	Envía el correo utilizando el protocolo smtp	

Clase: Options.java		
Descripción	Maneja la configuración de usuario y contraseña	
Atributo	Descripción	Tipo dato
imapAuthName	Campo para recibir el nombre de usuario	TextField
imapAuthPass	Campo para recibir la contraseña	TextField
smtpServer	Campo para recibir el servidor	TextField
smtpAuthName	Campo para recibir el nombre de usuario para conexión smtp	TextField
smtpAuthpass	Campo para recibir la contraseña para conexión smtp	TextField
Ok	Botón para aceptar la configuración realizada	Command
cancel	Botón para cancelar la configuración	command
Métodos	Características	
options	Constructor de la clase	

Clase: MailConnection.java		
Descripción	Crea una conexión con el protocolo imap o smtp	
Atributo	Descripción	Tipo dato
StreamConnection	Defina las capacidades de una conexión por ráfagas	StreamConnection
CRLF	Se asegura de que la comunicación con el protocolo sea línea por línea.	String
outputStream	Crea un flujo de bytes de salida	outputStream
inputStream	Crea un flujo de bytes de entrada	inputStream
b	Recibe la cadena proveniente del protocolo	Stringbuffer
ch	Contiene el valor decimal del carácter	Int
Métodos	Características	
mailConnection	Constructor de la clase	
getLine	Recibe las respuestas del protocolo	
send	Se utiliza para enviar cadenas al protocolo en forma de bytes	
sendCRLF	Añade el atributo CRLF al método send	
close	Cierra la conexión streamConnection	

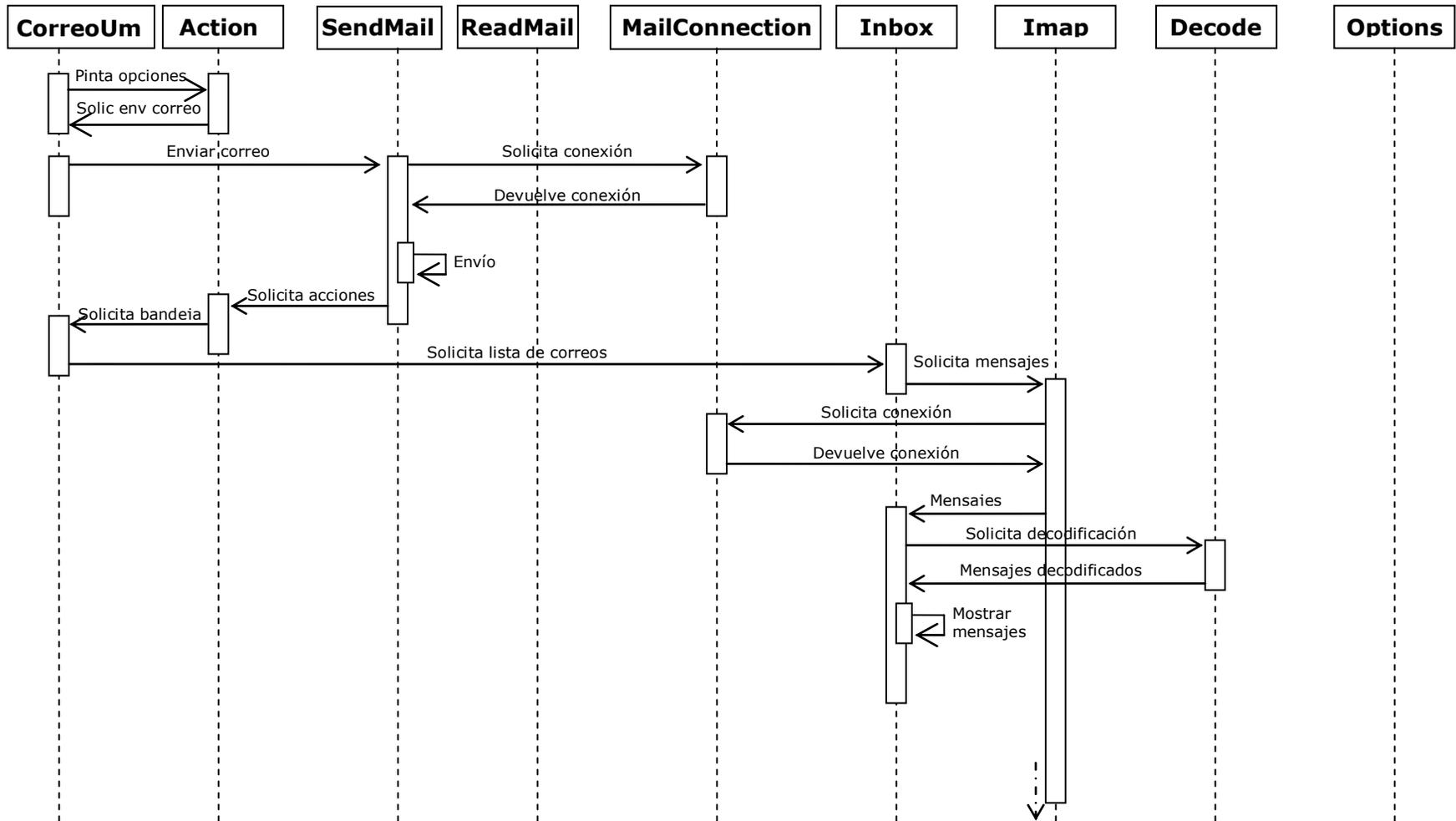
Clase: Imap.java		
Descripción	Hace todas las operaciones con el protocolo imap	
Atributo	Descripción	Tipo dato
Imap	Posee las características de la conexión	MailConnection
Reply	Recibe las respuestas del protocolo	String
Métodos	Características	
Imap	Constructor de la clase	
Open	Abre la conexión, autentica y solicita el buzón de mensajes	
numMessages	Extrae la cantidad de mensajes	
getHeader	Solicita el encabezado	
removeMessages	Hace el borrado real de los mensajes	
getBody	Solicita el cuerpo del mensaje	
getLine	Captura las respuestas del protocolo	
Close	Cierra la conexión	

Clase: Inbox.java		
Descripción	Se encarga de la gestión de la bandeja de entrada	
Atributo	Descripción	Tipo dato
Read	Comando para leer un correo	Command
Remove	Comando para borrar un correo	Command
exit	Comando para salir de la interfaz	Command
Imap	Instancia de la clase imap	Imap
Messages	Representa un mensaje en la bandeja de entrada	Vector
From	Almacena el origen de cada correo	String
Subject	Almacena el asunto de cada correo	String
Line	Recibe las líneas del protocolo suministradas por el objeto imap	String
Body2	Almacena el cuerpo de cada correo	String
Métodos	Características	
Inbox	Constructor de la clase, genera la interfaz de la bandeja de entrada	
readMails	Pone cada correo en la bandeja de entrada	
parseHeader	Da formato al encabezado	
readMail	Recorre cada correo para mostrarlo	
removeMail	Elimina cada correo	
Close	Cierra la interfaz	

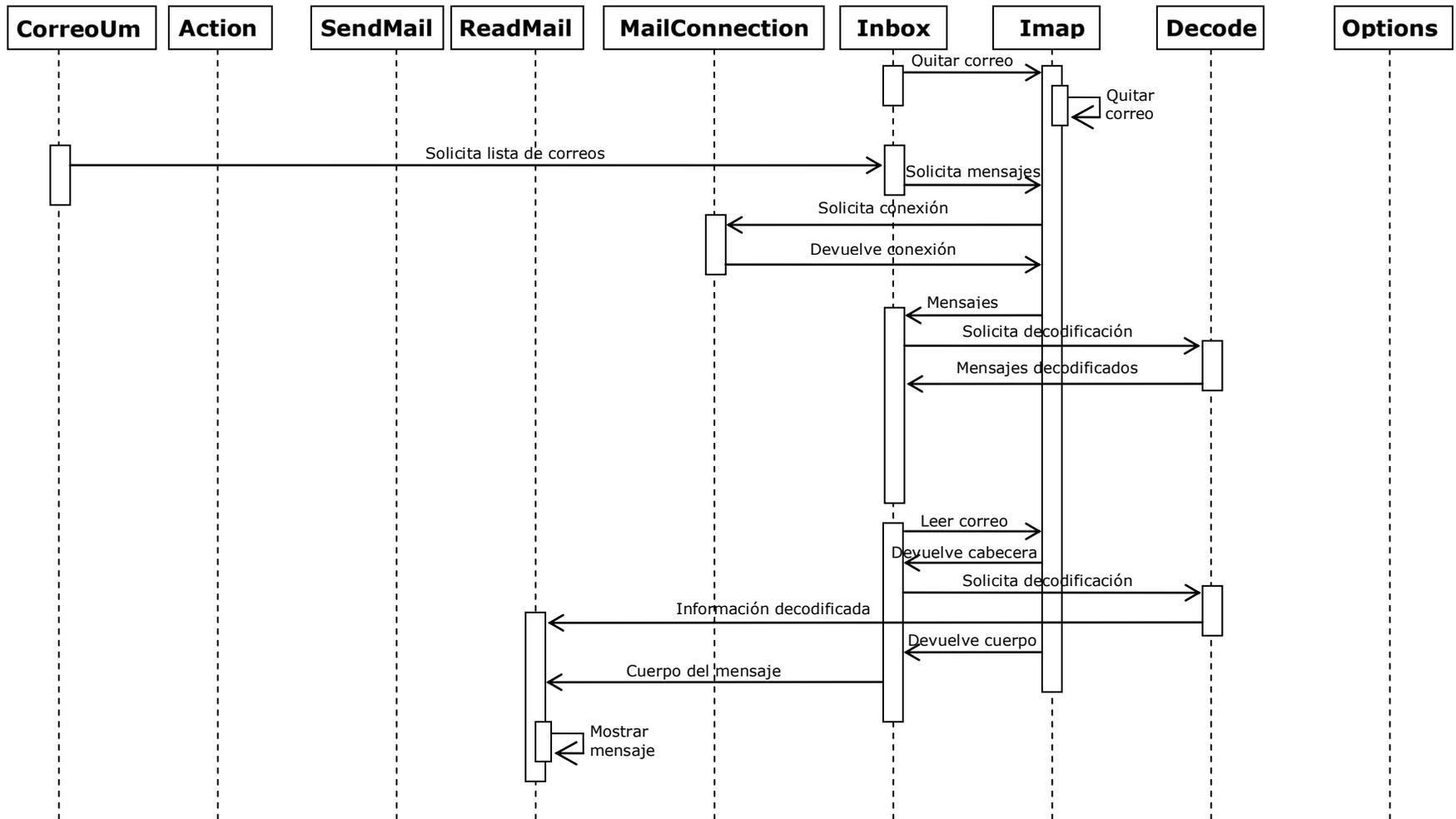
Clase: CorreoUM.java		
Descripción	Maneja la aplicación (clase principal)	
Atributo	Descripción	Tipo dato
Options	Instancia de la clase Options	Options
Action	Instancia de la clase Action	Action
Inbox	Instancia de la clase Inbox	Inbox
Mail	Instancia de la clase ReadMail	ReadMail
lastInbox	Identifica el último mensaje seleccionado	Int
sendMail	Instancia de la clase sendMail	sendMail
myDisplay	Instancia de Display para el manejo de pantalla y dispositivos de entrada	Display
Db	Almacena la información de configuración	RecordStore
From	Recibe el origen del correo	String
imapServer	Recibe el servidor imap	String
imapAuthName	Recibe el nombre de usuario para el protocolo imap	String
imapAuthPass	Recibe la contraseña para el protocolo imap	String
smtpServer	Recibe el servidor smtp	String
smtpAuth	Identifica si hay autenticación smtp o no	Byte
smtpAuthName	Recibe el nombre de usuario para el protocolo smtp	String
smtpAuthPass	Recibe la contraseña para el protocolo smtp	String
actualMessage	Almacena la posición del mensaje actual	Int
sendType	Identifica si el mensaje se envía, reenvía o responde	Int
Métodos	Características	

CorreoUM	Constructor de la clase
Run	Inicia la aplicación, pide lectura de correos
loadSettings	Carga una configuración guardada
saveSettings	Guarda una nueva configuración
startApp	Si existe una configuración muestra la interfaz Action
pauseApp	Para la aplicación
destroyApp	Destruye la aplicación
commandOptions	Edita los campos de usuario y la contraseña y muestra la interfaz Options
commandAction	Define las acciones de los comandos

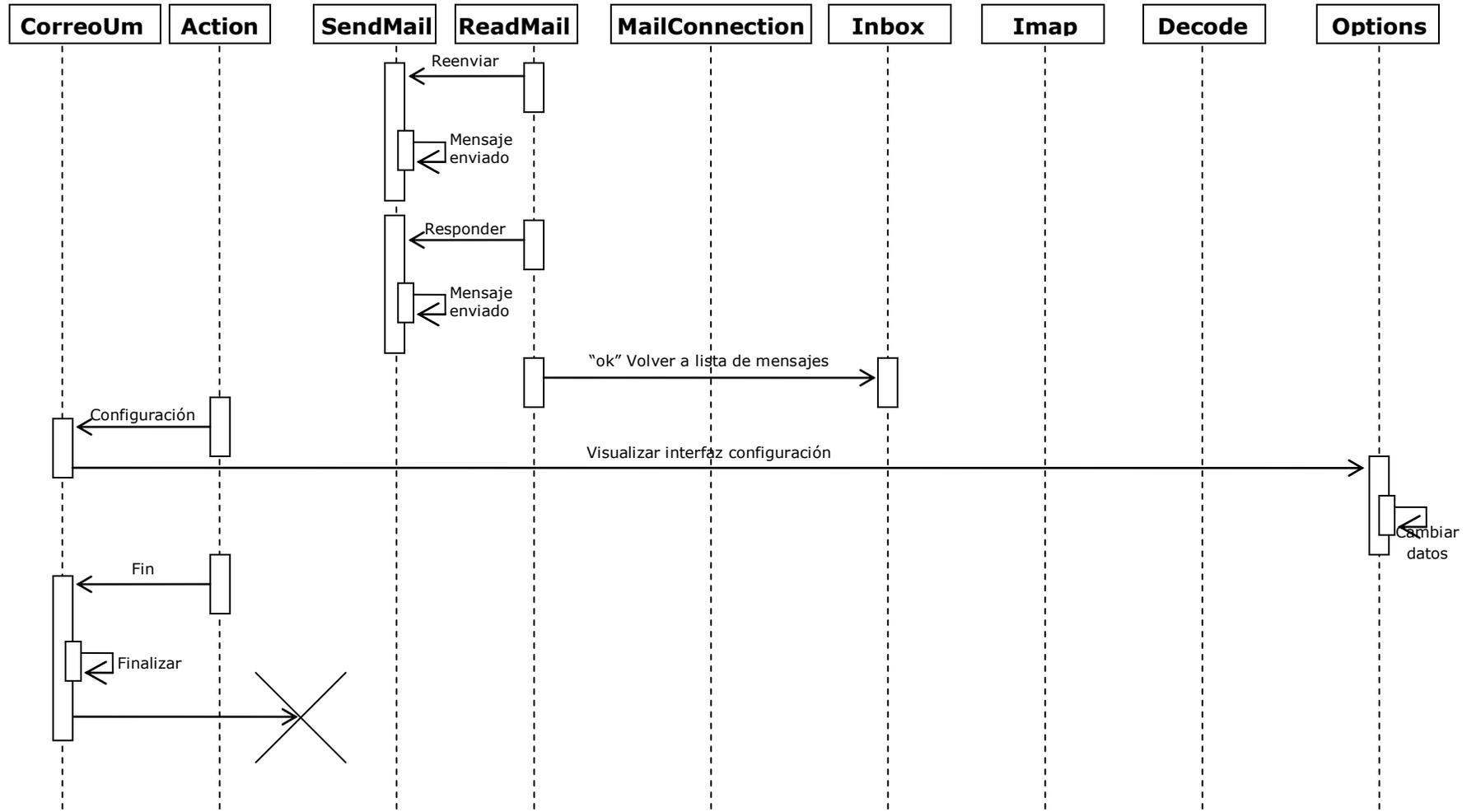
2.2 Diagrama de secuencia



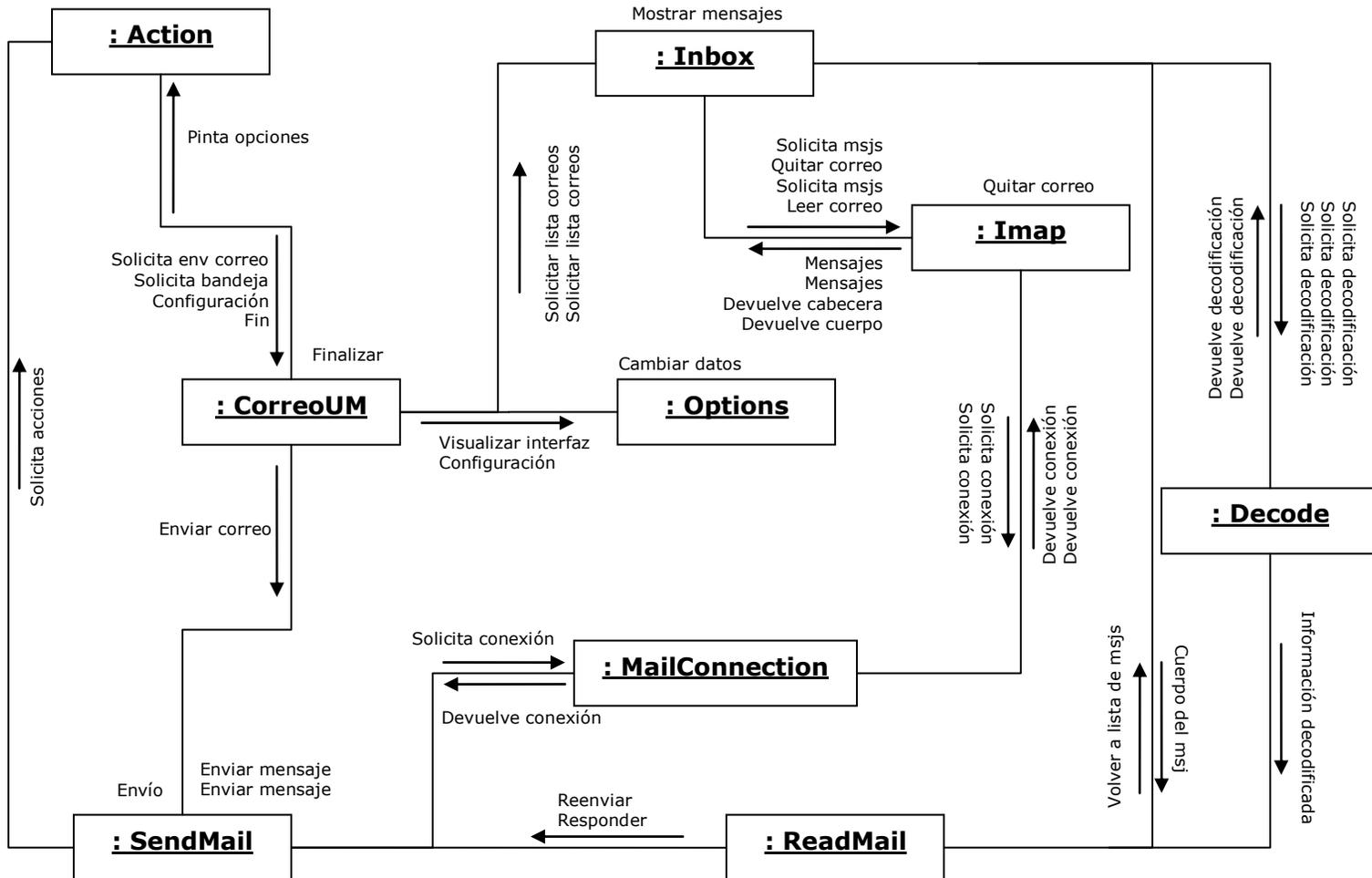
Continuación Diagrama de secuencia



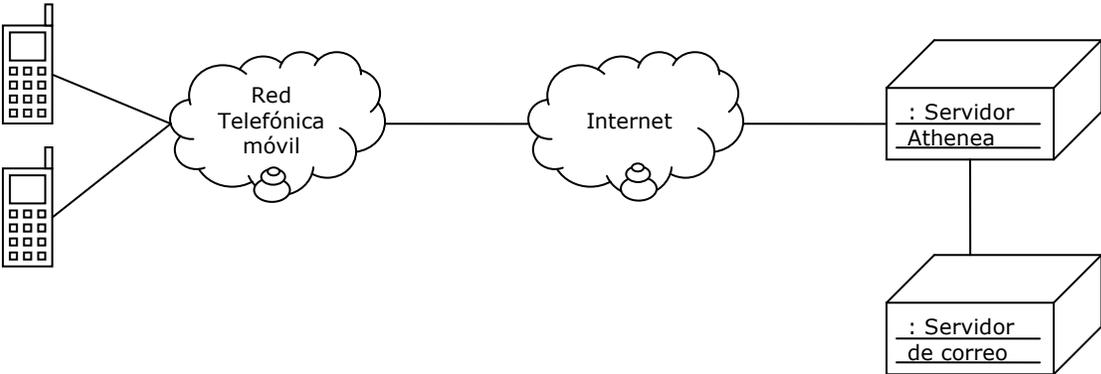
Continuación Diagrama de secuencia



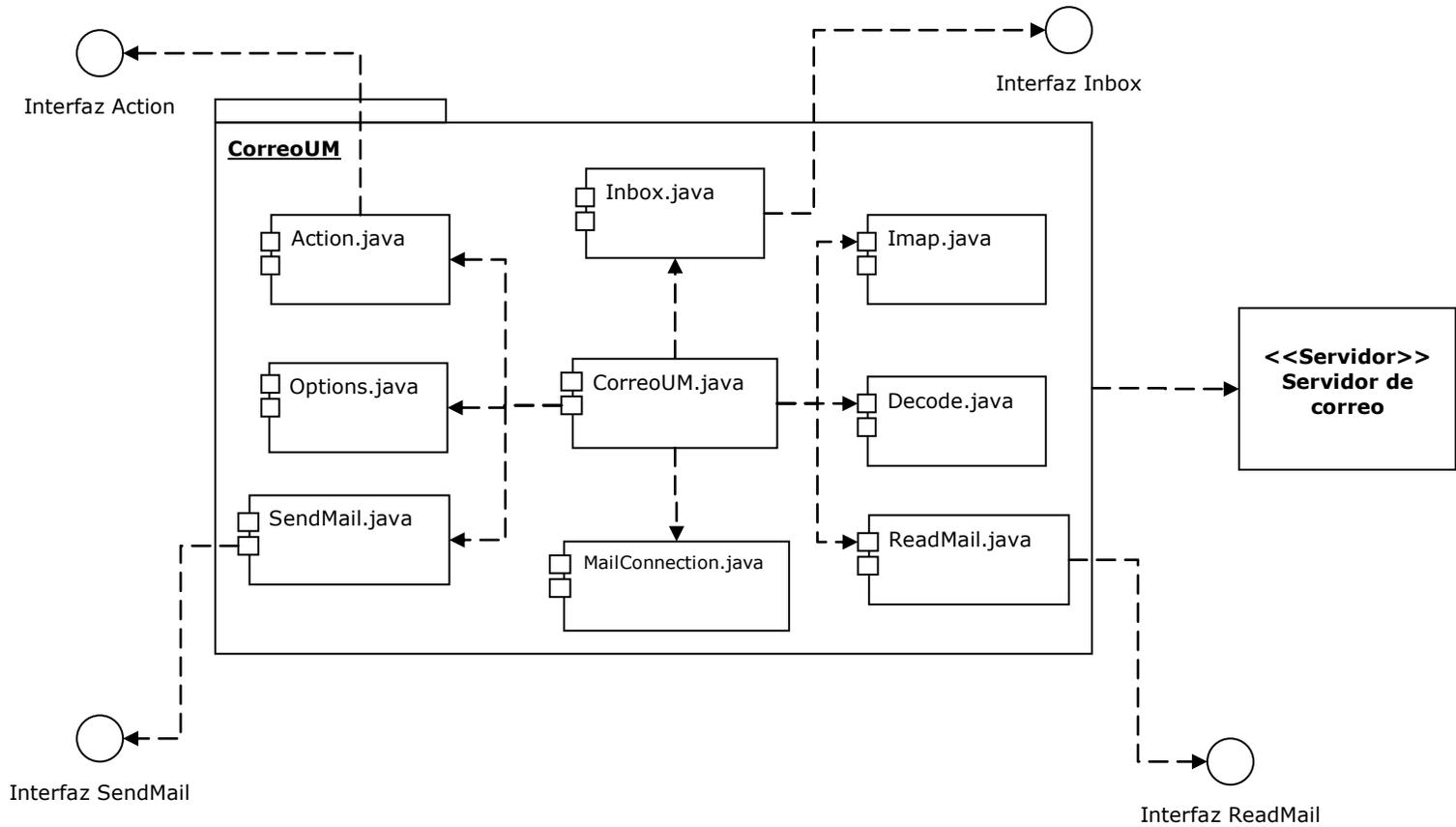
2.3 Diagrama de Colaboración



2.4 Diagrama de despliegue



2.5 Diagrama de componentes



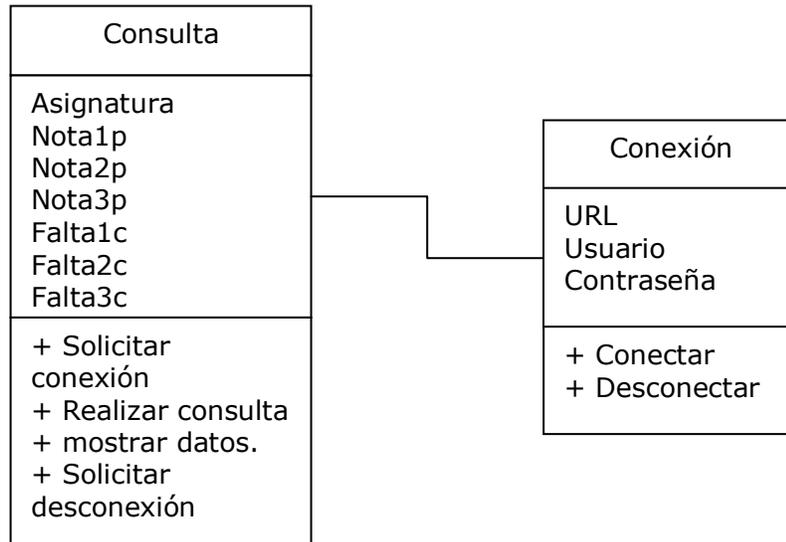
ANEXO C

ANÁLISIS Y DISEÑO DEL MÓDULO DE CONSULTA CONSULTAUM

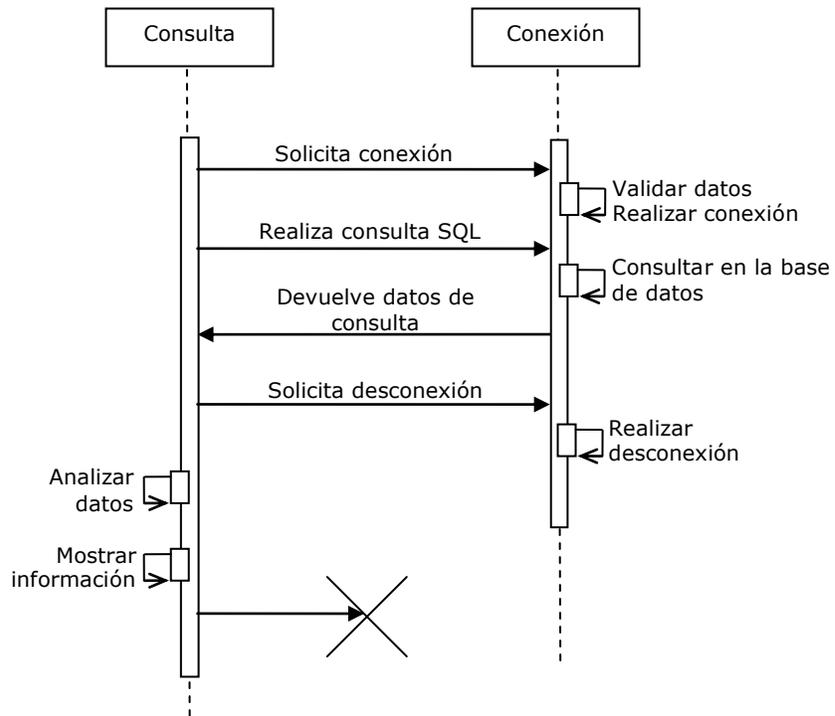
Descripción: este módulo permite realizar la consulta de notas y faltas de asistencia de las asignaturas inscritas por un estudiante en el semestre que esté cursando en el momento de realizar la consulta. El módulo utiliza un servicio Web para la conexión con la base de datos de Athenea.

1. Análisis Módulo de consulta ConsultaUM

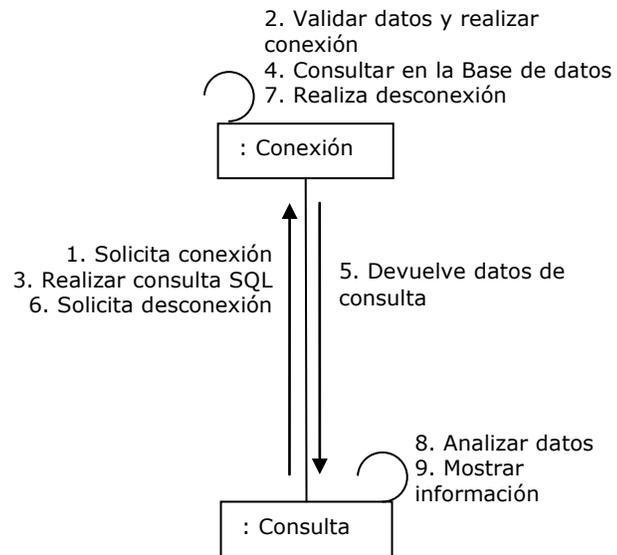
1.1 Diagrama de clases



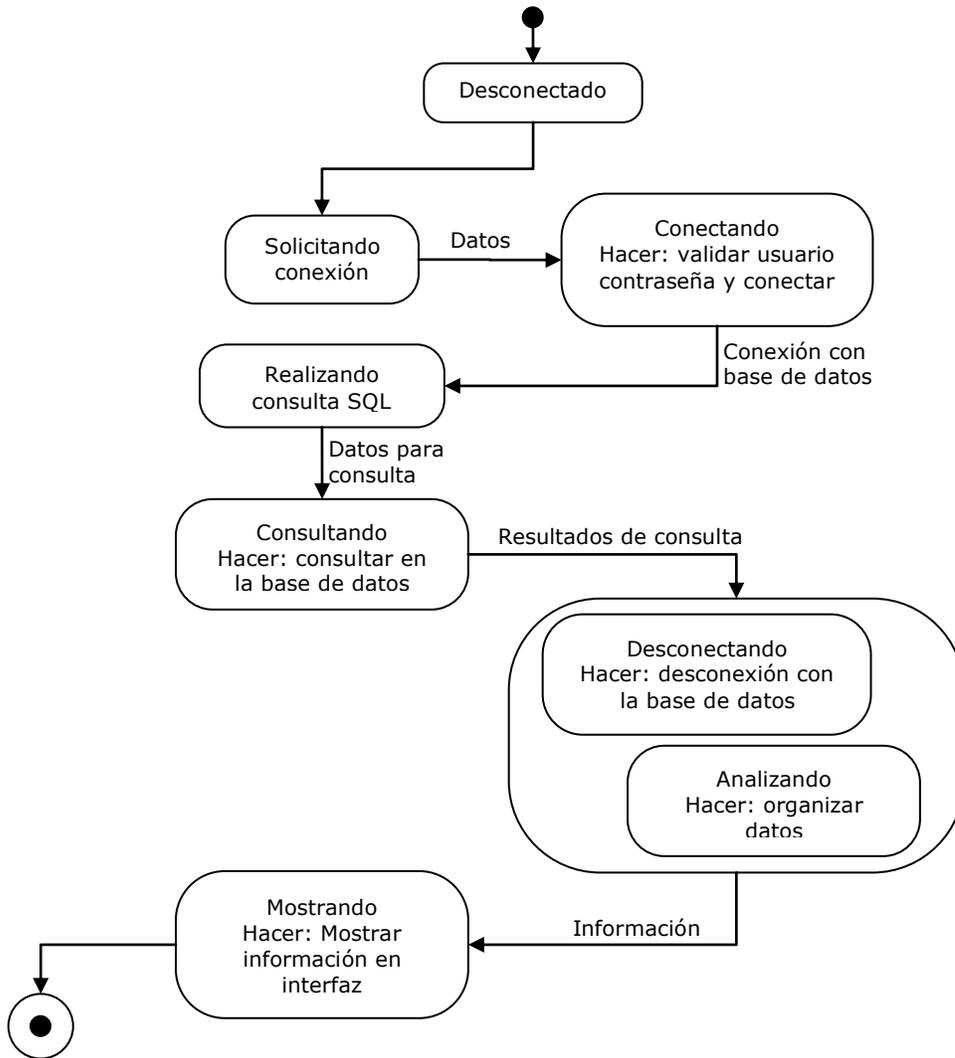
1.2 Diagrama de sucesos



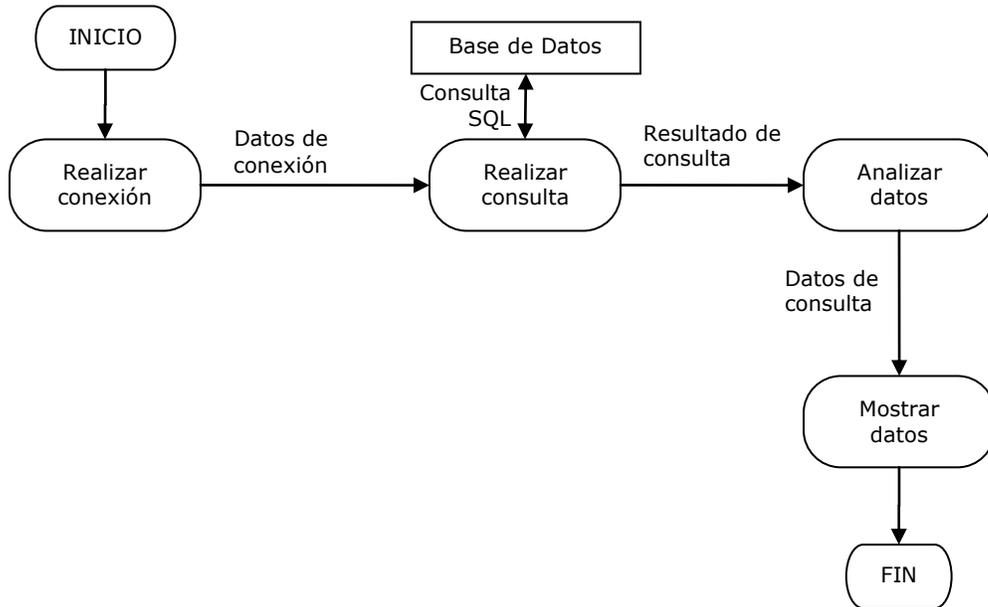
1.3 Diagrama de colaboración



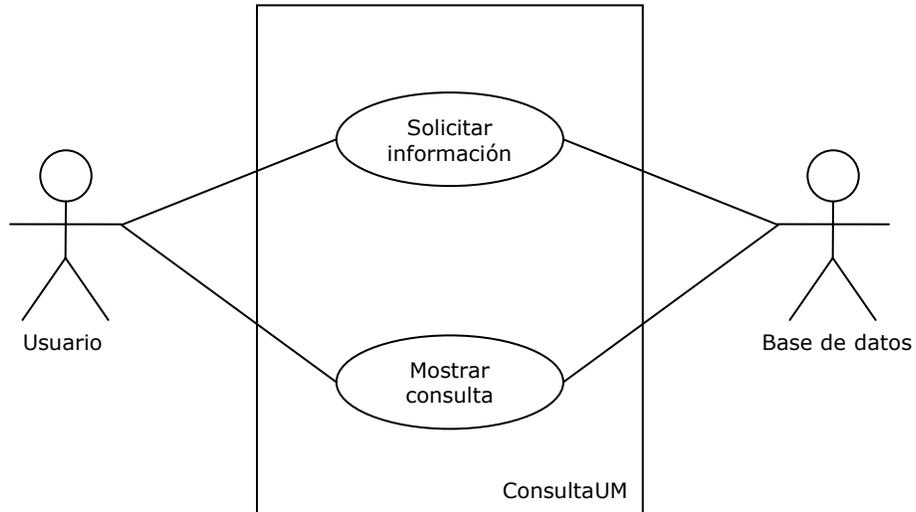
1.4 Diagrama de estados



1.5 Diagrama de flujo



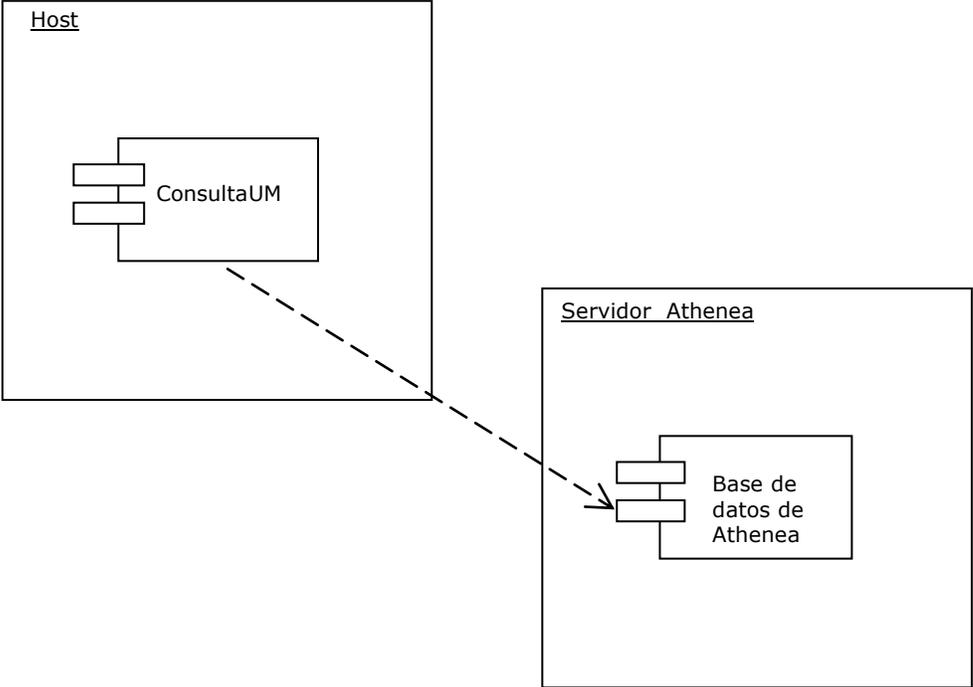
1.6 Diagrama de Casos de Uso



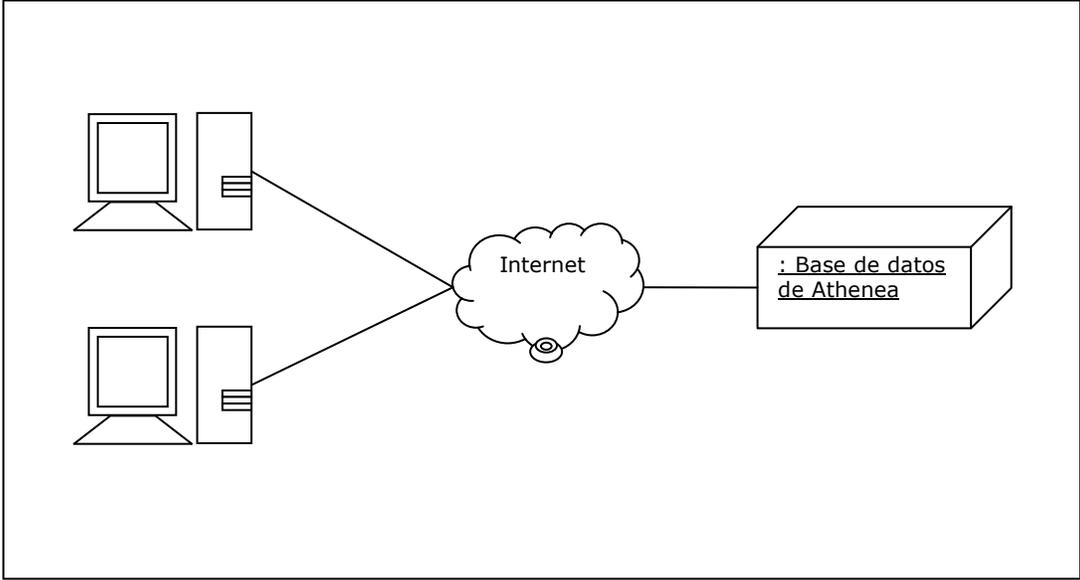
Caso de uso: Solicitar información	
Actores	Usuario, Base de datos.
Tipo	Primario.
Descripción	El usuario solicita la información de sus notas y asistencia.
Caso de uso anterior	

Caso de uso: Mostrar consulta	
Actores	Usuario, Base de datos.
Tipo	Primario.
Descripción	La base de datos devuelve a la aplicación los datos, para que los muestre al usuario de forma clara.
Caso de uso anterior	Solicitar información.

1.7 Diagrama de componentes

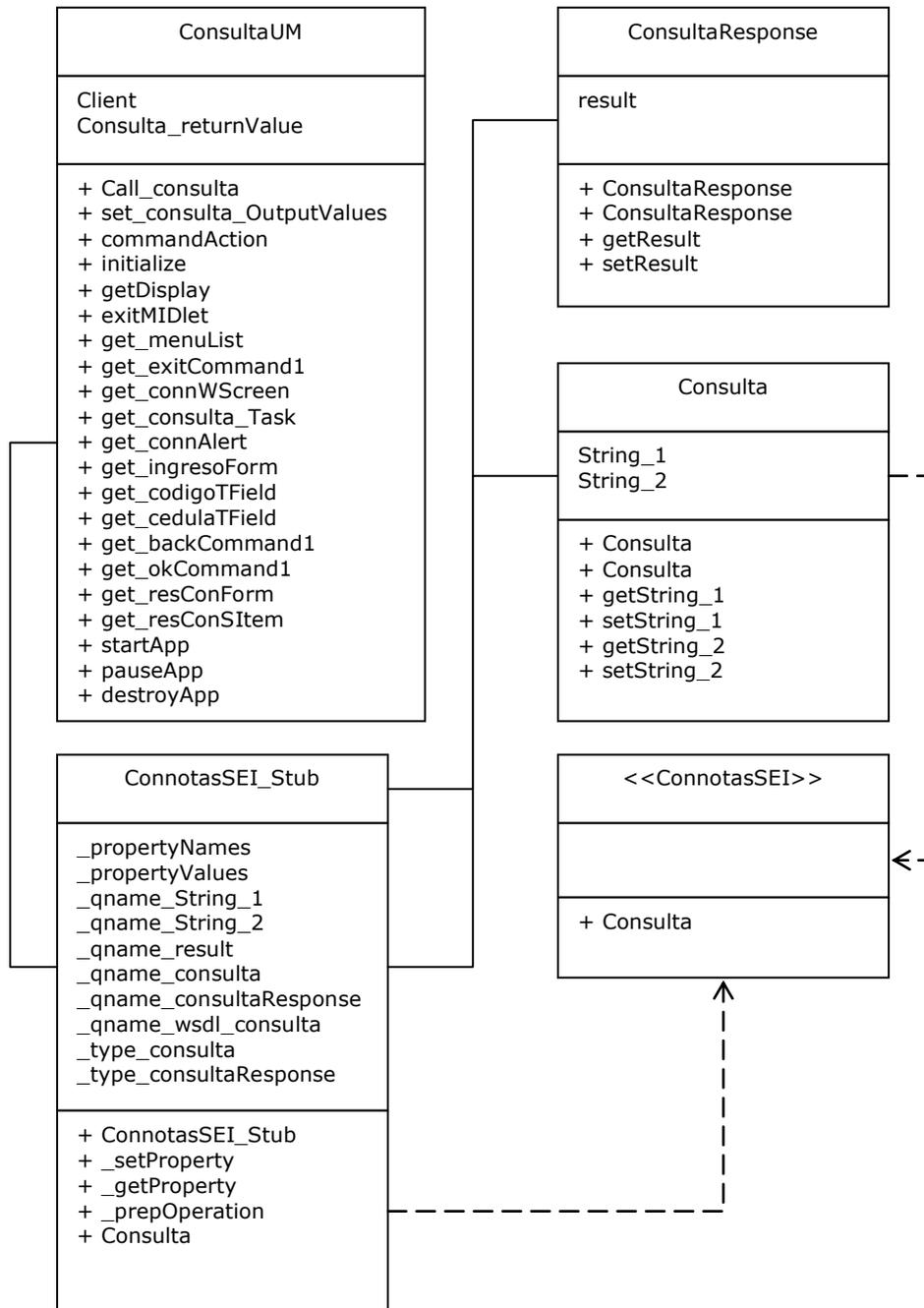


1.8 Diagrama de despliegue



2. Diseño Módulo de consulta de notas ConsultaUM

2.1 Diagrama de Clases



2.1.1 Diccionario de datos

Clase: ConsultaUM.java		
Descripción:	Es el MIDlet que controla la aplicación ConsultaUM.	
Atributo	Descripción	Tipo dato
Client	Representa el objeto servidor del lado del cliente.	ConnotasSEI_Stub
Consulta_returnValue	Campo para recibir el valor retornado por el método consulta.	String
Métodos	Características	
Call_consulta	El cliente hace el llamado al servicio.	
Set_consulta_OutputValues	Recibe los valores de la consulta.	
commandAction	Indica al sistema que comando ha sido invocado en cualquier pantalla.	
Initialize	Inicializa la interfaz de usuario.	
getDisplay	Retorna una instancia de la pantalla.	
exitMIDlet	Termina la aplicación.	
Get_menuList	Retorna una instancia de menuList componente de netbeans que representa una lista.	
Get_exitCommand1	Retorna una instancia del comando de netbeans exitCommand1	
get_connWScreen	Retorna una instancia de connWScreen, componente de netbeans que representa una pantalla de espera.	
get_consulta_Task	Retorna una instancia de consulta_Task, componente de netbeans que representa una tarea de cancelación simple, en este caso, una consulta.	
get_connAlert	Retorna una instancia de connAlert, componente de netbeans que representa una pantalla de mensaje de alerta.	
get_ingresoForm	Retorna una instancia de ingresoForm, que representa el componente Formulario de netbeans.	
get_codigoTField	Retorna una instancia de codigoTField que representa un campo de texto de netbeans.	
get_cedulaTField	Retorna una instancia de cedulaTField que representa un campo de texto de netbeans.	
get_backCommand1	Retorna una instancia de backCommand1, un comando de netbeans que regresa a la pantalla anterior.	
get_okCommand1	Retorna una instancia del comando de netbeans okCommand1, que representa la afirmación de una acción.	
get_resConForm	Retorna una instancia de resConForm, un componente de netbeans que representa un formulario para mostrar los datos de la consulta.	
get_resConSItem	Retorna una instancia de resConSItem, un componente de netbeans que representa un String para mostrar los datos.	
startApp	Inicia la aplicación.	
pauseApp	Pausa la aplicación.	
destroyApp	Termina la aplicación.	

Clase: ConnotasSEI_Stub.java	
Descripción:	Genera el esquema xml a través del cual se hace uso del servicio.

Atributo	Descripción	Tipo dato
_propertyNames	Representa el nombre de la propiedad, en este caso la propiedad de dirección de punto final, que contiene la dirección en la que se encuentra el servicio Web.	String
_propertyValues	Representa el valor de la propiedad anterior.	Object
_qname_String_1	Nombre cualificado que representa un nombre de atributo	QName
_qname_String_2	Nombre cualificado que representa un nombre de atributo	QName
_qname_result	Nombre cualificado que representa un nombre de atributo	QName
_qname_consulta	Nombre cualificado que representa un nombre de atributo	QName
_qname_consultaResponse	Nombre cualificado que representa un nombre de atributo	QName
_qname_wsdl_consulta	Nombre cualificado que representa un nombre de atributo	QName
_type_consulta	Tipo de elemento	Element
_type_consultaResponse	Tipo de elemento	Element
Métodos	Características	
ConnotasSEI_Stub	Constructor de la clase	
_setProperty	Define las propiedades name y value	
_getProperty	Obtiene las propiedades name y value	
_prepOperation	Adjunta las variables a la operación	
consulta	Ejecuta la operación consulta	

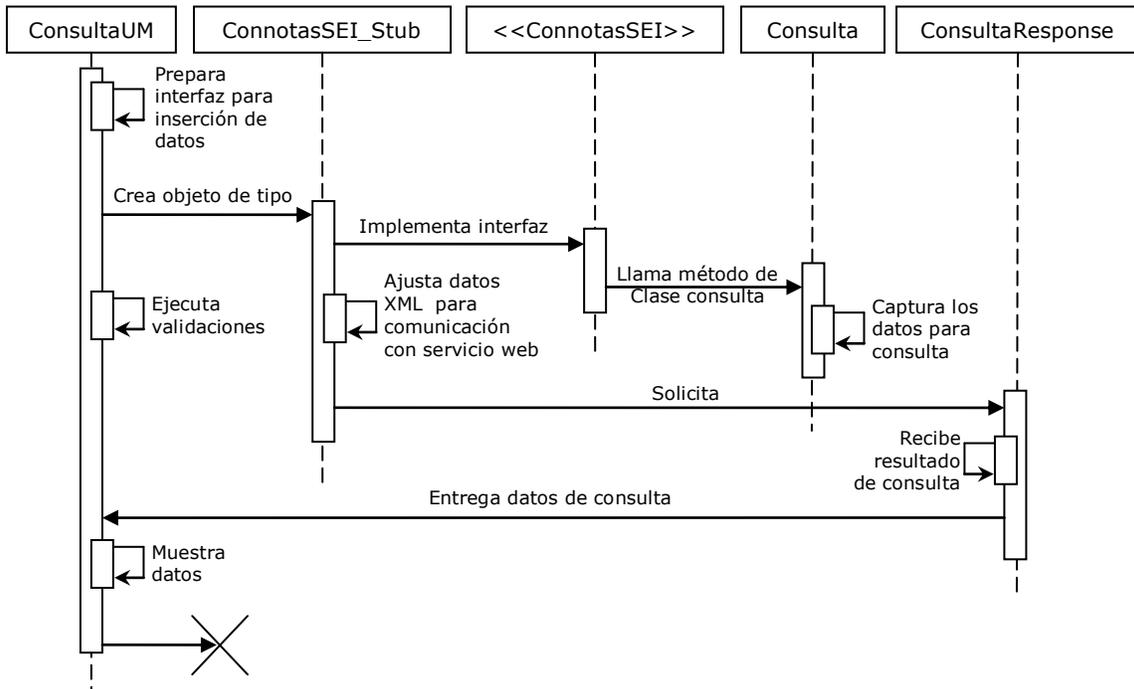
Clase: ConsultaResponse.java		
Descripción:	Clase que recibe el resultado de la consulta	
Atributo	Descripción	Tipo dato
Result	Resultado de la consulta	String
Métodos	Características	
ConsultaResponse	Constructor de la clase.	
ConsultaResponse	Crea una instancia de result, un elemento protegido de tipo String.	
getResult	Obtiene el valor de la variable result	
setResult	Adiciona el valor de la variable result	

Clase: Consulta.java		
Descripción:	Gestiona las variables para la consulta (código, cédula)	
Atributo	Descripción	Tipo dato
String_1	Código del estudiante a consultar	String
String_2	Cédula del estudiante a consultar	String
Métodos	Características	
Consulta	Constructor de la clase	
Consulta	Realiza una instanciación de la variables String_1 y String_2	

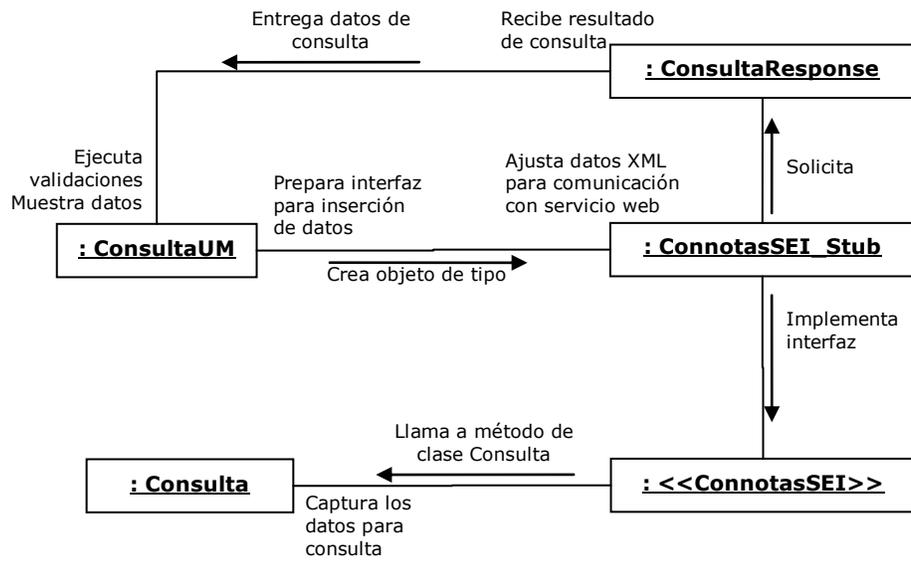
getString_1	Obtiene el valor de la variable String_1
setString_1	Adiciona el valor a la variable String_1
getString_2	Obtiene el valor de la variable String_2
setString_2	Adiciona el valor a la variable String_2

Interfaz: ConnotasSEI.java		
Descripción:	Carga la ejecución del proceso remoto (consulta)	
Atributo	Descripción	Tipo dato
Métodos	Características	
Consulta	Representa el método consulta que se encuentra en el servidor.	

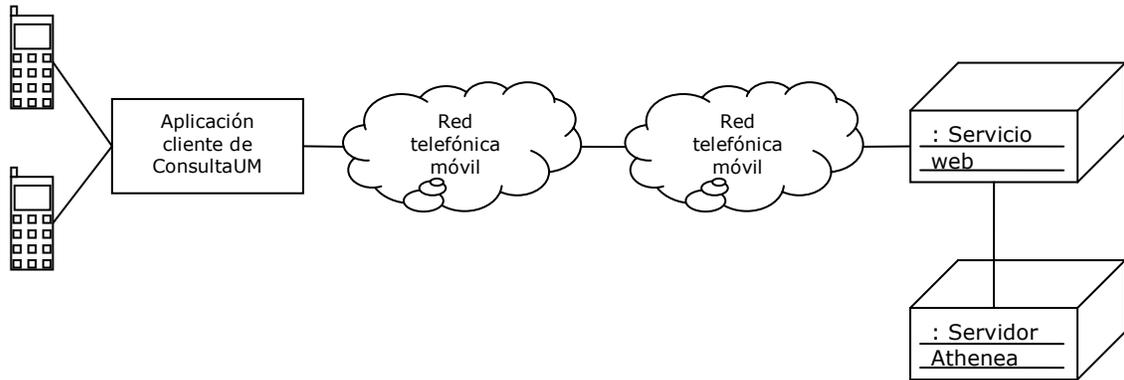
2.2 Diagrama de secuencia



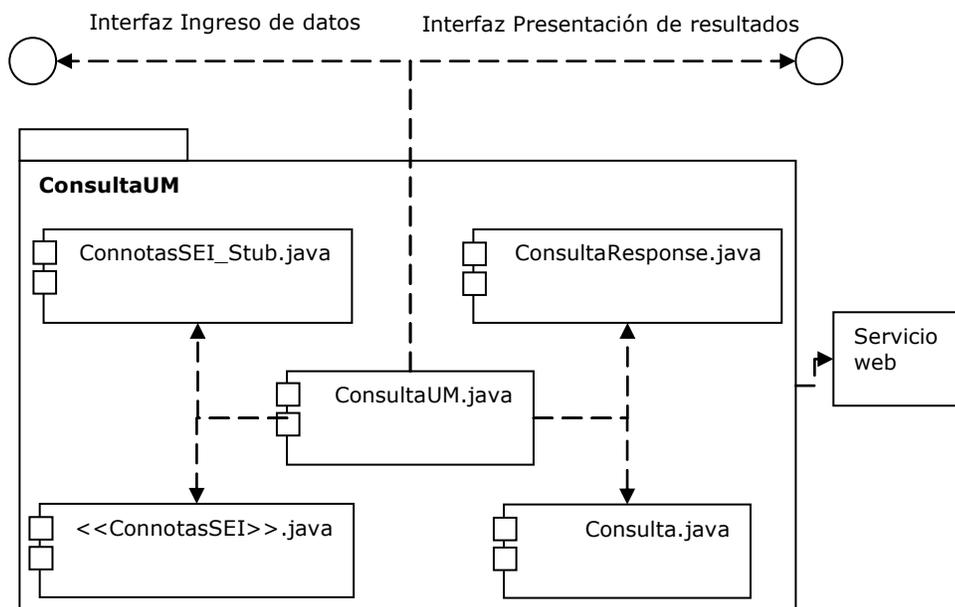
2.3 Diagrama de colaboración



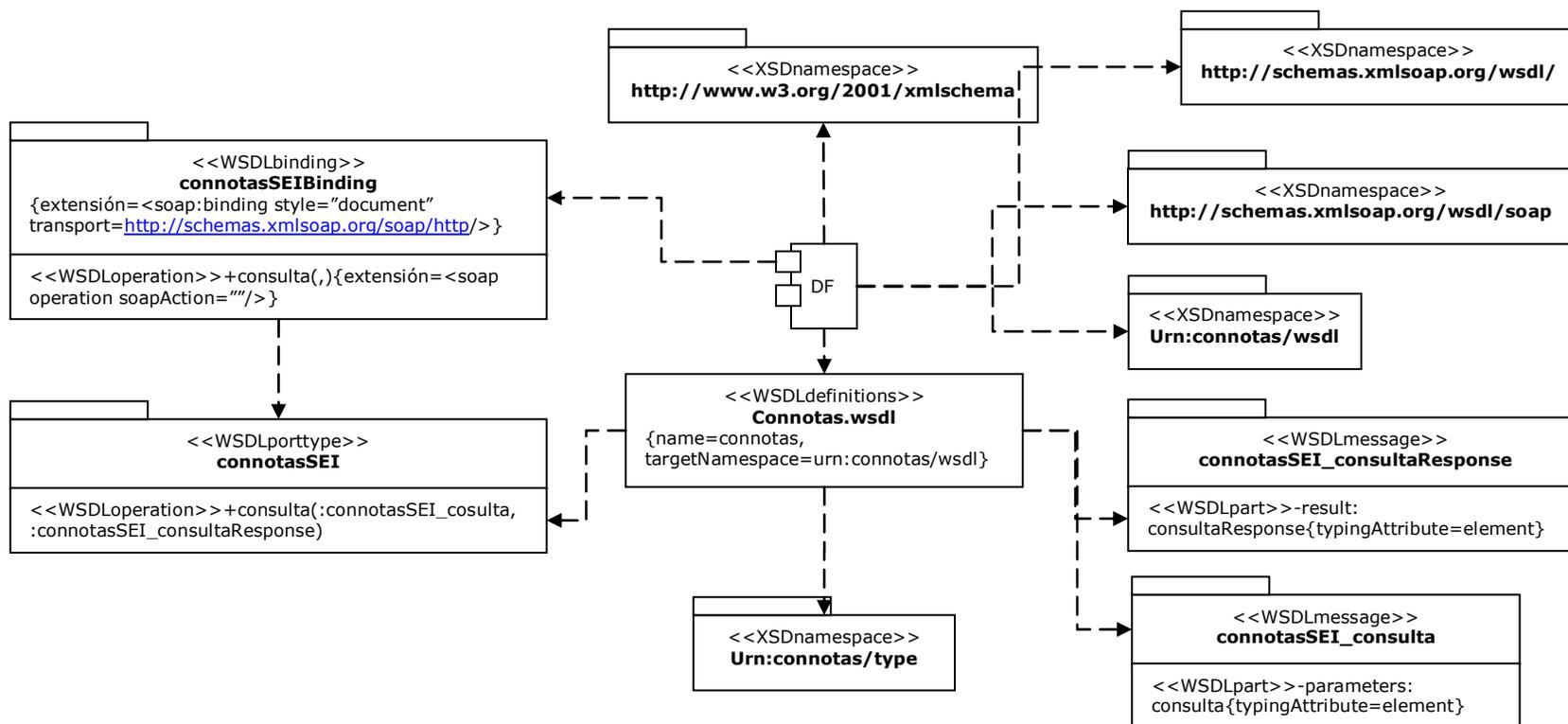
2.4 Diagrama de despliegue



2.5 Diagrama de componentes



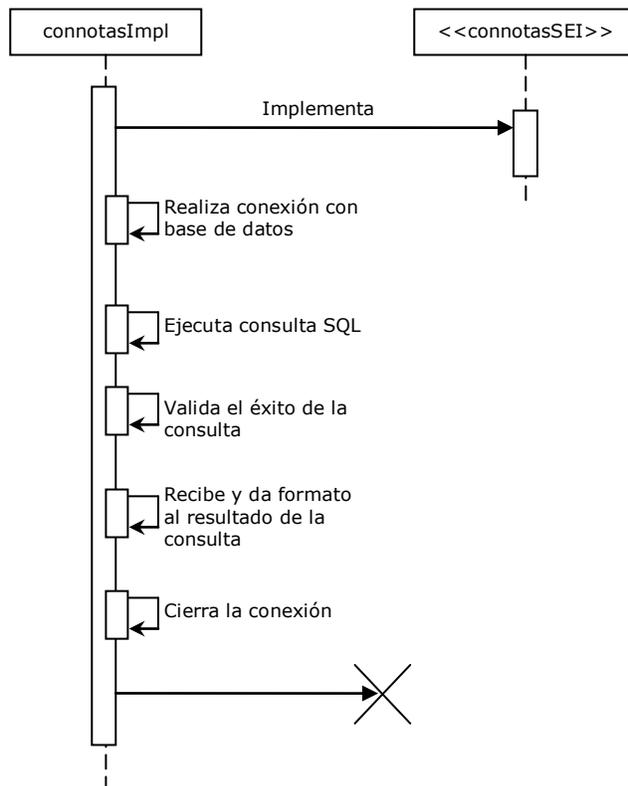
2.6 Diagrama WSDL para el Servicio WEB



2.7 Diagrama de clases para el Servicio WEB



2.8 Diagrama de secuencia para el servicio WEB



ANEXO D

MANUAL DE USUARIO

Descarga de las Aplicaciones desde la página Web

Para ingresar a la página Web, se debe acceder al navegador del dispositivo móvil y digitar la url: <http://glec.umanizales.edu.co:8080/Descargas/>. Si existe cualquier duda con este proceso remitirse al manual de usuario del dispositivo utilizado.

Al entrar a la página Web se encuentra la siguiente pantalla:



Para escoger la aplicación a descargar, se debe señalar con las teclas de dirección del teclado el enlace con su nombre y luego elegir la opción seleccionar; luego con el mismo botón se acepta la descarga y ya está.

CORREOUM

Ejecutar la Aplicación en el Dispositivo Móvil

En la carpeta en la que se encuentren las aplicaciones descargadas desde Internet se encuentra la aplicación correoUM (luego de ser descargada). Para ejecutar esta aplicación se debe seleccionar y/o ejecutar con el botón del celular que realice esta operación. En el ejemplo, la aplicación puede ser ejecutada presionando el botón central llamado *SELECT* o el botón que se encuentra al lado derecho que indica la ejecución de la acción *Launch*.



Ingresar Datos (Configuración)

Luego de ejecutarse la aplicación, aparece la primera vez que se ejecute en el dispositivo, una pantalla en la que se encuentran dos campos en los que se debe ingresar el nombre de usuario y la contraseña del usuario del servidor Athenea que esté utilizando la aplicación.

En el primer campo titulado *usuario* se debe ingresar el nombre de usuario para la cuenta de correo; en el segundo campo *clave* irá la contraseña de dicha cuenta.

Para aceptar los datos y continuar con los pasos de la aplicación, se deben aceptar los datos digitados; en el ejemplo se ve que el botón del lado superior derecho del teclado indica la acción *Aceptar*, que permite a la aplicación tomar los datos que se ingresaron.

Para volver a la pantalla anterior (Ejecutar la Aplicación en el Dispositivo Móvil), se debe presionar el botón que indica la acción *Cancelar*, que en el ejemplo es el botón superior izquierdo del teclado.



Menú de opciones (Opciones)

Luego de ingresar el nombre de usuario y la contraseña en la pantalla anterior (configuración), se accede a las opciones generales de la aplicación:

- Escribir mail: permite acceder a una nueva pantalla con las opciones necesarias para enviar un correo electrónico.
- Leer mail: accede a una pantalla en la que se muestra la bandeja de entrada de la cuenta de correo del usuario.
- Configuración: permite volver a la pantalla en la que se ingresan los datos del usuario para la conexión a la cuenta.
- Salir: cierra la aplicación y se ubica en el lugar en el que está instalada

Para escoger una opción se debe seleccionar con las teclas de dirección del dispositivo móvil y luego elegir la opción seleccionada; en el caso del ejemplo, las teclas de dirección (arriba, abajo, izquierda, derecha) se ubican en el botón superior central del teclado, y presionando éste en el centro se ejecuta la acción *Seleccionar*.



Escribir Mail

En esta opción del menú como su nombre lo indica se puede escribir un mensaje para enviarlo a cualquier destinatario de correo electrónico.

Como se muestra en la siguiente figura existen tres campos en el primero de ellos, *Para:* se debe digitar la dirección de correo electrónico a la que se desea enviar el mensaje, la siguientes son: *Asunto:* y *Contenido:* que llevan la información que su nombre indica.



Oprimiendo el botón superior izquierdo del teclado se ejecuta la acción enviar, ésta finaliza la edición del mensaje y manda el correo a su destinatario, a su vez el comando cancelar que se encuentra en el botón superior derecho del teclado, cancela la edición, el envío y regresa a la pantalla anterior (Menú de Opciones).

CONSULTAUM

Ejecutar la aplicación en el dispositivo móvil

Para ejecutar la aplicación de consulta de notas, debe seleccionarse en la pantalla en la que se muestran las aplicaciones descargadas al dispositivo.

Una vez seleccionada la aplicación se debe presionar el botón que indica la acción *Launch*, que se encuentra en el botón superior derecho en el ejemplo; también se puede abrir utilizando el botón que indique la selección, en este caso *SELECT*. Luego debe seleccionarse el método que trae la aplicación llamado también consulta; este se ejecuta presionando el botón seleccionar, *SELECT* en el ejemplo. Para salir se debe presionar *Exit*.



Ingresar los datos

Luego de haber seleccionado el método consulta de la aplicación, aparece una interfaz en la que se encuentran dos campos de ingreso de datos. El primero es el campo *Código* en el que se debe ingresar el código del estudiante; el segundo campo *cédula* es el campo en el que se digita el número del documento de identidad del usuario.

Para salir esta interfaz y volver a la anterior, se debe presionar el botón que ejecuta la acción *Back* que se encuentra en el botón superior izquierdo del teclado del dispositivo del ejemplo. Para aceptar el ingreso de datos y continuar con el proceso de consulta se debe presionar el botón que ejecute la opción *Ok* que se encuentra, en el ejemplo, en el botón superior derecho.



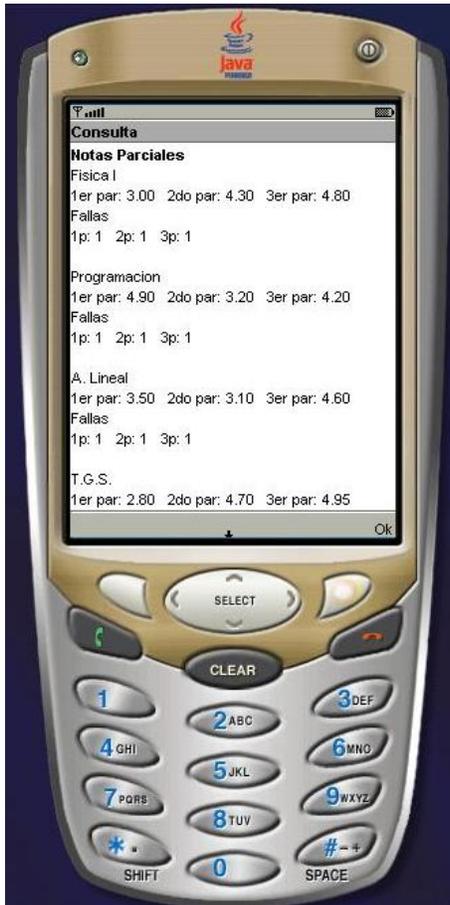
Conexión con la base de datos

Si los datos ingresados en los campos *código* y *cédula* son correctos, la aplicación muestra una interfaz con la leyenda *Obteniendo Datos* que aparece mientras la aplicación se conecta con el servicio web y accede a los datos. Si la información ingresada en estos campos no es correcta, la aplicación mostrará una interfaz en la que aparece un mensaje de error. Para terminar este mensaje debe presionarse el botón que realice la acción *Done* que en el ejemplo, se encuentra en el botón superior derecho del teclado. Con esto se consigue volver a la interfaz en la que se podrán ingresar unos datos válidos para la consulta.



Visualización de la información

Una vez la aplicación se ha conectado con el servicio Web, que a su vez realiza la conexión con la base de datos, y que tiene la información lista, la interfaz con el mensaje *Obteniendo Datos* finaliza y aparece una interfaz que tiene por título *Consulta* y que muestra la información discriminada de las notas de las asignaturas que el estudiante tenga inscritas en el semestre que esté cursando. Esta información está detallada por materias y cortes (primer parcial, segundo parcial y tercer parcial o final); además la consulta muestra al usuario la información de las fallas por cada materia para cada uno de los cortes o periodos por parcial (1p, 2p, 3p).



La información que se presenta en esta interfaz puede superar la capacidad de la pantalla, por lo tanto, una flecha indicado mas información aparecerá. Para ver la información que no se está presentando todavía debe presionarse el botón de dirección *abajo* con lo cual se consigue revisar el resto de información que aun no se había visto.

Para terminar, debe presionarse el botón que indica la ejecución de la acción *Ok* y que permite volver al método *consulta* de la aplicación.

LECTOR RSSUM

Introducción

La aplicación RssUM cuenta con dos utilidades, la primera permite leer los feeds o alimentadores que se encuentren en la lista, que por defecto tiene solamente el feed de Athenea llamado en la aplicación, Athenea RSS.

La otra utilidad es un editor de feeds que permite añadir elementos a la lista de del lector RSS.

Estas dos aplicaciones están disponibles luego de haber descargado el módulo de lectura RssUM de la página Web; para utilizarlas debe seleccionarse una de ellas con los botones de dirección *arriba/abajo* que en este caso se encuentran rodeando al botón *SELECT* que está ubicado en la parte superior central del teclado, luego presionando el botón que indique la acción seleccionar, que en el caso del ejemplo se encuentra en el botón superior central del teclado *SELECT* o la opción *Launch* ubicada en el botón superior derecho del teclado.



El lector RssUM



Este lector permite leer feeds o alimentadores de una lista predeterminada, que por defecto cuenta con el feed de Athenea solamente.

Al seleccionarlo aparece una pantalla temporal en la que se muestran las dos aplicaciones del módulo y el logo de la Universidad de Manizales.

Cuando se cambia la lista de feeds con el editor de feeds (explicado mas adelante), se pueden encontrar en ella varios elementos que pueden ser seleccionados; para ello deben utilizarse los botones que realizan la acción de dirección *arriba/abajo*, que en el ejemplo están en el botón superior central del teclado.

Cuando la lista se encuentra en su estado original, solo aparece un elemento, el feed Athenea RSS, que permite acceder a los titulares del portal Athenea. Este elemento aparece seleccionado por defecto y para acceder

a él debe presionarse el botón seleccionar que se encuentra en la parte superior central del teclado en el ejemplo, **SELECT**.

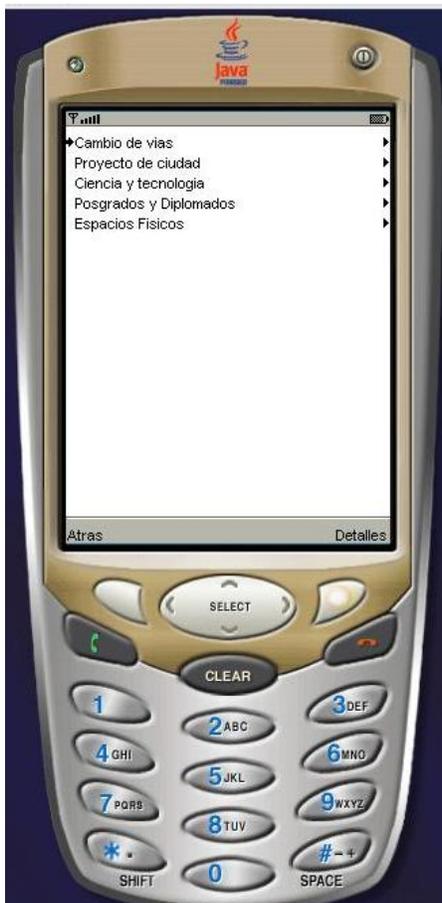


Titulares

Si la conexión con el feed no se puede realizar, un mensaje de error aparecerá, indicando que no se pudo conectar y recomendando intentarlo nuevamente o tratar con otro feed de la lista, si se encuentran varios elementos en ella.

En la pantalla aparece una lista de titulares del alimentador seleccionado cuando la conexión se puede realizar con éxito.

Para seleccionar un titular se deben utilizar los botones de dirección *arriba/abajo* que se encuentran en el botón superior central del teclado en el ejemplo. Cuando un elemento está seleccionado aparece una flecha al lado izquierdo de la pantalla que apunta al titular escogido.



Cada titular cuenta con una imagen al lado derecho del mismo; la imagen es un triángulo que apunta hacia la derecha indicando que para desplegar la información del titular seleccionado debe presionarse el botón de dirección *derecha* que en el ejemplo se encuentra en la parte superior central del teclado.

Otra forma de desplegar la información que presenta un titular es presionando en botón que ejecuta la acción *Detalles* que se encuentra en el botón superior derecho del teclado en el ejemplo.

Para volver a la ventana anterior, en la que se encuentra la lista de alimentadores del lector se debe presionar la opción *Atrás*, indicada en el botón superior izquierdo del ejemplo.



Titulares desplegados

Cuando se despliega un titular aparece en la pantalla el titular en la parte superior, separado del cuerpo de éste por una línea punteada.

Al lado izquierdo del titular aparece un triángulo que apunta hacia la izquierda indicando que, para volver a la pantalla anterior en la que se encuentran los titulares, debe presionarse el botón de dirección *izquierda* que se encuentra, en el ejemplo, en el botón superior central del teclado.

Para volver a la pantalla anterior, se puede también presionar el botón que ejecuta la acción indicada *Back* que se encuentra en el botón superior izquierdo del teclado en el ejemplo.



El editor

La otra aplicación del módulo es el editor de feeds o alimentadores, que permite añadir o eliminar elementos de la lista que utiliza el lector.

En la pantalla inicial de la utilidad aparecen los feeds que se encuentran en la lista de la aplicación. Para seleccionar uno de los elementos de la lista se utilizan los botones de dirección *arriba/abajo* que se encuentran, en el ejemplo, en el botón superior central del teclado.

Para salir de la aplicación se debe presionar el botón que indica la acción *Salir* que está en el botón superior izquierdo del teclado en el ejemplo.

Seleccionando la opción *Menu* que se encuentra en la parte inferior derecha de la pantalla, se accede al menú de opciones del editor. En



el se puede escoger entre crear un nuevo feed *nuevo*, borrar un feed *borrar* y dejar la lista en su estado original *por defecto*. Para seleccionarlas deben utilizarse los botones *arriba/abajo* y luego seleccionar *SELECT*, botón que se encuentra en la parte superior central del teclado.



Utilizando la opción borrar se pueden eliminar elementos de la lista, uno a uno. Cuando se utiliza la opción *Por defecto*, se elimina cualquier feed que se haya creado y queda solamente el feed de Athenea.

Creación de un feed

Cuando se utiliza la opción *Nuevo* del menú del editor de feeds, aparece una nueva pantalla en la que se encuentran dos campos de ingreso de datos.

En el campo llamado *Nombre* se debe ingresar el nombre del nuevo feed. Este campo aparece seleccionado por defecto y se puede digitar la información inmediatamente aparece la interfaz. La información debe ingresarse con las letras que se indican en cada uno de los números del teclado, como se indique en el manual de usuario del dispositivo.

Luego de escribir el nombre del nuevo feed y para digitar la dirección de éste; se debe utilizar el botón de dirección *abajo* para seleccionar el campo de ingreso llamado *URL*.



En este campo debe colocarse la ruta del alimentador en la red; por ejemplo, si la ruta es www.a.com/rss.xml en el campo debe colocarse esta con el prefijo *http://*, así

<http://www.a.com/rss.xml>. La aplicación valida el ingreso de los datos y no permite que se añadan feeds con el mismo nombre de cualquiera que esté en la lista, ni se pueden añadir feeds sin el prefijo <http://>. Cuando los datos no sean ingresados adecuadamente, un mensaje de error aparecerá; para salir de este mensaje debe seleccionarse la opción *Done* que ejecuta el botón superior derecho del teclado en el ejemplo.



Anexo E

Manual Técnico

Gestiones con el servidor

Para la instalación de la página Web de descarga de aplicaciones en el servidor GLEC se debe instalar primero el Sun Java System Application Server, lo cual debe hacerse con las opciones por defecto de la instalación, luego debe ingresarse a la página de administración del servidor, en el caso del GLEC es: <http://glec.umanizales.edu.co:4848/> e ingresar el nombre de usuario y contraseña definidos en la instalación. Ingresar al enlace *Tareas Comunes*, allí seleccionar el enlace Implementar *aplicación Web (.war)*, luego seleccionar el radio button que dice “Especifique un archivo empaquetado para cargarlo en Application Server.” Dar click en el botón examinar, para buscar la ubicación del archivo WAR de la página; luego dar click en el botón *siguiente* y después en el botón *aceptar*.

Para la configuración del servicio web del módulo consultaUM, se deben seguir los pasos mencionados en el párrafo anterior, solo se que se debe seleccionar el archivo WAR del servicio.

Luego de esto se debe reiniciar el servidor con los comandos:

```
asadmin stop-domain (se baja el servidor)
asadmin start-domain domain1 (se levanta el servidor)
```

El paso anterior es para actualizar los cambios en el servidor.

Gestiones con la base de datos

Primero debe crearse el usuario *GLEC*:

En una consola, como usuario root:

```
root# useradd GLEC
root# passwd GLEC (seguir las indicaciones en la consola para ingresar el password)
```

Como usuario postgres, dentro de la base de datos eval_ver2:

```
postgres# CREATE USER GLEC ENCRYPTED PASSWORD 'contraseña';
```

Luego debe autorizarse el acceso (para consulta) a las tablas *inscmate*, *materias* y *estudiantes*:

```
postgres# GRANT select ON inscmate, materias, estudiantes TO GLEC;
```

Para darle acceso al host que contiene el web service, a la base de datos del servidor (PostgreSQL), se debe modificar el archivo *pg_hba.conf* de la siguiente manera:

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
```

```
# IPv4 local connections:
```

```
Host eval_ver2 GLEC 200.75.46.1 2255.255.255.255 md5
```

Para que el servicio Web pueda consultar la base de datos en postgres, debe agregarse el driver postgresql-8.1-408.jdbc3.jar en JAVA_HOME\jre\lib\ext\

REQUERIMIENTOS:

El módulo de consulta solo corre sobre dispositivos móviles cuya virtual machine soporta web services, es decir, que tengan la versión MIDP 2.0 y CLDC 1.1.