

ESCUELA POLITÉCNICA DEL EJÉRCITO

SEDE LATACUNGA



CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

Desarrollo de una aplicación que permita la integración de telefonía celular a través de mensajes de texto como medio de acceso a la información para la consulta de notas de los alumnos de la Escuela Politécnica del Ejército Sede Latacunga.

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS

FAUSTO ANDRÉS ARMENDARIZ VALDIVIESO

Latacunga, abril 2010

ESCUELA POLITÉCNICA DEL EJÉRCITO
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Yo, Fausto Andrés Armendariz Valdivieso

Autorizo a la Escuela Politécnica del Ejército la publicación, en la biblioteca virtual de la institución del trabajo “DESARROLLO DE UNA APLICACIÓN QUE PERMITA LA INTEGRACIÓN DE TELEFONÍA CELULAR A TRAVÉS DE MENSAJES DE TEXTO COMO MEDIO DE ACCESO A LA INFORMACIÓN PARA LA CONSULTA DE NOTAS DE LOS ALUMNOS DE LA ESCUELA POLITÉCNICA DEL EJÉRCITO SEDE LATACUNGA.” Cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Latacunga, abril del 2010

Fausto Andrés Armendariz Valdivieso

C.I. 0501896286

ESCUELA POLITÉCNICA DEL EJÉRCITO
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

DECLARACIÓN DE RESPONSABILIDAD

Yo, Fausto Andrés Armendariz Valdivieso

DECLARO QUE:

El proyecto de grado denominado “DESARROLLO DE UNA APLICACIÓN QUE PERMITA LA INTEGRACIÓN DE TELEFONÍA CELULAR A TRAVÉS DE MENSAJES DE TEXTO COMO MEDIO DE ACCESO A LA INFORMACIÓN PARA LA CONSULTA DE NOTAS DE LOS ALUMNOS DE LA ESCUELA POLITÉCNICA DEL EJÉRCITO SEDE LATACUNGA.” Ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, abril del 2010

Fausto Andrés Armendariz Valdivieso

C.I. 0501896286

ESCUELA POLITÉCNICA DEL EJÉRCITO
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

CERTIFICADO

Que el trabajo titulado “DESARROLLO DE UNA APLICACIÓN QUE PERMITA LA INTEGRACIÓN DE TELEFONÍA CELULAR A TRAVÉS DE MENSAJES DE TEXTO COMO MEDIO DE ACCESO A LA INFORMACIÓN PARA LA CONSULTA DE NOTAS DE LOS ALUMNOS DE LA ESCUELA POLITÉCNICA DEL EJÉRCITO SEDE LATACUNGA. realizado por el señor Fausto Andrés Armendariz Valdivieso ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el reglamento de estudiantes de la Escuela Politécnica del Ejército.

Debido a que constituye un trabajo de excelente contenido científico que coadyuvará a la aplicación de conocimientos y al desarrollo profesional, si recomiendan su publicación

Ing. Santiago Jácome

DIRECTOR DE PROYECTO

Ing. César Naranjo

CODIRECTOR DE PROYECTO

AGRADECIMIENTOS

En la vida una persona se esfuerza por cumplir las metas que se traza, pero existen piedras en el camino hacia esas metas, piedras que nos hacen tropezar y que muchas veces podremos levantarnos solos, pero muchas más necesitaremos de la ayuda de alguien que nos tienda una mano para poder seguir éste camino.

En mi caso estas personas han estado siempre a mi lado durante todo el camino, y no solo para ayudarme a levantar ante cualquier tropiezo, sino también para advertirme de las piedras que vendrán en el camino y para alentarme a seguir siempre adelante sin temer a las adversidades, sin importar cuán pedregoso sea el camino.

Ojala las palabras plasmadas en este papel pudieran tan solo expresar la mitad de este sentimiento de gratitud hacia las personas que siempre están incondicionalmente apoyándome, ya que no creo que exista palabra ni frase que demuestre tan grande sentimiento.

Estas personas son mis hermanos, tías y padres que siempre han estado a mi lado vigilando por mi bienestar, apoyándome ante cualquier adversidad.

Fausto Armendariz

DEDICATORIA

Este trabajo, mis estudios y todo lo que hago está dedicado a mis padres que me han apoyado en todo momento de mi vida, que son mí inspiración para superarme, que cada día me enseñan a ser mejor, que hacen que mi vida sea cada día mejor.

A mis tías por su capacidad de dar todo por mí, sin esperar nada a cambio, por amarme sin condiciones, aceptando mis errores, festejando mis logros, sufriendo mis dolores.

Fausto Armendariz

TABLA DE CONTENIDOS

CONTENIDO	PÁGS.
Portada	i
Autorización	ii
Declaración de responsabilidad	iii
Certificación	iv
Agradecimientos	v
Dedicatoria	vi
Tabla de contenidos	vii-xv
Resumen	xvi

CAPÍTULO I LA TECNOLOGÍA CELULAR

1.1 INTRODUCCIÓN	1-2
1.2 HISTORIA Y EVOLUCIÓN	2-4
1.2.1 GENERACIONES DE LA TELEFONÍA CELULAR	4
1.2.1.1 Generación Cero (0G).....	4
1.2.1.2 Primera generación (1G).....	4-5
1.2.1.3 Segunda generación (2G).....	5
1.2.1.4 Generación 2.5 G.....	5-6
1.2.1.5 Tercera generación (3G).....	6
1.3 TECNOLOGÍAS DE ACCESO CELULAR	7-8
1.4 ESTÁNDARES	8
1.4.1 AMPS	8-9

1.4.2 GSM.....	9-10
1.5 FABRICANTES.....	10
1.5.1 NOKIA.....	11
1.5.2 SAMSUNG.....	11
1.5.3 MOTOROLA.....	11
1.5.4 LG ELECTRONICS.....	11-12
1.5.5 SONY ERICSSON MOBILE COMMUNICATIONS AB.....	12
1.5.6 APPLE INC.....	12
1.6 EMPRESAS EN ECUADOR.....	12
1.6.1 MOVISTAR.....	12-13
1.6.2 PORTA.....	13-14
1.6.3 ALEGRO PCS.....	14
1.7 MENSAJES DE TEXTO (SMS).....	
1.7.1 ASPECTOS IMPORTANTES	
1.7.1.1 Definición.....	15
1.7.1.2 Historia.....	15-16
1.7.1.3 Funcionamiento de los SMS.....	16-17
1.7.1.4 Ventajas del SMS.....	17
1.7.2 COMPONENTES.....	18
1.7.3 FORMATOS.....	18
1.7.3.1 SMS En Formato Texto.....	18
1.7.3.2 SMS En Formato PDU.....	18-19
<i>1.7.3.2.1 Estructura de los SMS en formato PDU.....</i>	<i>19-22</i>
1.7.4 ESTÁNDARES.....	22

1.7.4.1 GSM	22-23
1.7.4.2 CDMA2000	23
1.7.4.3 UMTS	23-24
1.7.5 COSTOS DEL SERVICIO	25-26

CAPÍTULO II CONECTIVIDAD ENTRE EL PC Y EL TELÉFONO CELULAR

2.1 PUERTOS DE COMUNICACIONES	27
2.1.1 PUERTOS PS/2	
2.1.1.1 Definición.....	27
2.1.1.2 Características.....	27-28
2.1.1.3 Forma.....	28
2.1.1.4 Ubicación en el sistema informático.....	28
2.1.2 PUERTOS USB (UNIVERSAL SERIAL BUS)	
2.1.2.1 Definición.....	28-29
2.1.2.2 Características.....	29
2.1.2.3 Forma.....	29
2.1.2.4 Ubicación en el sistema informático.....	30
2.1.3 PUERTOS SERIALES (COM)	
2.1.3.1 Definición.....	30
2.1.3.2 Características.....	30-31
2.1.3.3 Forma.....	31
2.1.3.4 Ubicación en el sistema informático.....	31
2.1.4 PUERTOS RJ-45	

2.1.4.1 Definición.....	31
2.1.4.2 Características.....	31
2.1.4.3 Forma.....	32
2.1.4.4 Ubicación en el sistema informático.....	32
2.2 CONEXIÓN ENTRE EL CELULAR Y EL PC.....	32
2.2.1 CABLE USB.....	32
2.2.2 BLUETOOTH.....	33
2.2.2.1 Descripción.....	33
2.2.2.2 Especificaciones.....	34-35
2.2.3 ENLACE INFRARROJO.....	35
2.2.3.1 Descripción.....	35-36
2.2.3.2 Características.....	36
2.3 EL CELULAR COMO MÓDEM EXTERNO.....	36
2.3.1 MÓDEM.....	37
2.3.1.1 Cómo funciona?.....	37
2.3.1.2 Módems para PC.....	37-39
CAPÍTULO III ESTUDIO DE HERRAMIENTAS, EQUIPOS Y DISPOSITIVOS EMPLEADOS	
3.1 INTRODUCCIÓN.....	40
3.2 ESTUDIO DE HERRAMIENTAS DE DESARROLLO	40-47
3.3 ECLIPSE.....	47
3.3.1 ARQUITECTURA.....	47-48
3.4 ESTUDIO DE EQUIPOS Y DISPOSITIVOS	

A UTILIZARSE	48-50
3.5 ESTUDIO DE SISTEMAS OPERATIVOS Y	
DRIVERS	50
3.5.1 SISTEMA OPERATIVO	50-52
3.5.2 DRIVERS	52
CAPÍTULO IV DESARROLLO DE LA	
APLICACIÓN	
4.1 DESCRIPCIÓN DE LA APLICACIÓN	53
4.2 ESPECIFICACIÓN DE REQUISITOS	
4.2.1 INTRODUCCIÓN	54
4.2.1.1 Propósito.....	54
4.2.1.2 Ámbito de la aplicación.....	54
4.2.1.3 Definición de acrónimos y abreviaturas.....	
4.2.1.3.1 <i>Definiciones</i>	55
4.2.1.3.2 <i>Acrónimos</i>	55
4.2.1.3.3 Abreviaturas.....	55
4.2.1.4 Referencias.....	55
4.2.1.5 Visión general del documento.....	55
4.2.2 DESCRIPCIÓN GENERAL	56
4.2.2.1 Perspectiva del producto.....	56
4.2.2.1.1 <i>Funciones de la aplicación</i>	56-57
4.2.2.2 Características de los usuarios.....	57
4.2.2.3 Restricciones.....	57
4.2.2.3.1 <i>Suposiciones</i>	58

4.2.2.3.2 Dependencias.....	58
4.2.3 REQUISITOS ESPECÍFICOS.....	58
4.2.3.1 Requisitos funcionales	
4.2.3.1.1 Conexión con un teléfono celular.....	58
4.2.3.1.2 Recibir mensajes del usuario.....	58
4.2.3.1.3 Descargar mensajes de la bandeja de entrada.....	58
4.2.3.1.4 Analizar el mensaje de texto.....	59
4.2.3.1.5 Realizar consulta en la base de datos.....	59
4.2.3.1.6 Estructurar un mensaje de texto.....	59
4.2.3.1.7 Registrar eventos.....	59
4.2.3.1.8 Enviar el mensaje al usuario.....	59
4.2.3.2 Requisitos de interfaces externas	
4.2.3.2.1 Interfaces de usuario.....	60
4.2.3.2.2 Interfaces de software.....	60
4.2.3.2.3 Interfaces de comunicación.....	60
4.2.3.3 Requisitos de rendimiento.....	60
4.2.3.4 Requisitos de desarrollo.....	60
4.2.3.5 Requisitos tecnológicos.....	60
4.2.3.6 Atributos.....	
4.2.3.6.1 Seguridad.....	60-61
4.3 MODELAMIENTO.....	61
4.3.1 DIAGRAMA DE CASOS DE USO	62
4.3.1.1 Definición de casos de uso de alto nivel.....	63-72
4.3.2 DIAGRAMAS DE ESTADO.....	72-76

4.3.3 DIAGRAMAS DE CLASES.....	77
4.4 IMPLEMENTACIÓN.....	78-99
5. CONCLUSIONES Y RECOMENDACIONES	
5.1 CONCLUSIONES.....	100-101
5.2 RECOMENDACIONES.....	101-102
ANEXOS.....	103-109
REFERENCIAS BIBLIOGRÁFICAS.....	110-112

ÍNDICE DE FIGURAS

FIGURA	PÁGS.
2.1 Puertos PS/2.....	28
2.2 Forma de puertos PS/2.....	28
2.3 Conectores USB.....	30
2.4 Conectores DB-9 macho y hembra.....	31
2.5 Conectores RJ-45.....	32

ÍNDICE DE TABLAS

TABLAS	PÁGS.
1.1 Estructura de un SMS enviado.....	19
1.2 Elementos TIPO PDU SMS enviado.....	19
1.3 Codificación a 7 bits.....	20
1.4 Transformación a octetos.....	20
1.5 Ejemplo de envío de SMS en formato PDU.....	20
1.6 Estructura de un SMS recibido.....	21

1.7 Elementos TIPO PDU SMS recibido.....	21
1.8 Estructura del tiempo de recepción SMS recibido.....	21
1.9 Costos SMS Movistar.....	25
1.10 Costos SMS Porta.....	25
1.11 Costos SMS Alegro.....	26
2.1 Clases de Bluetooth.....	34
2.2 Versiones de Bluetooth.....	34
3.1 Características del computador utilizado.....	49
3.2 Características del teléfono celular utilizado.....	49
3.3 Requerimientos mínimos Windows Vista.....	52
4.1 Definiciones.....	55
4.2 Acrónimos.....	55
4.3 Abreviaturas.....	55
4.4 Caso de uso recibir mensajes.....	63
4.5 Caso de uso conexión con el teléfono celular.....	64
4.6 Caso de uso descargar mensajes de la bandeja de entrada.....	66
4.7 Caso de uso analizar el mensaje de texto.....	67
4.8 Caso de uso consultar base de datos.....	68
4.9 Caso de uso registrar eventos.....	69
4.10 Caso de uso estructurar un mensaje de texto.....	70
4.11 Caso de uso enviar mensajes.....	71

ÍNDICE DE DIAGRAMAS

DIAGRAMAS	PÁGS.
4.1 Diagrama de casos de uso.....	62
4.2 Diagrama de secuencia recibir mensajes.....	63
4.3 Diagrama de secuencia conexión con el teléfono celular.....	65
4.4 Diagrama de secuencia descargar mensajes de la bandeja de entrada....	66
4.5 Diagrama de secuencia analizar el mensaje de texto.....	67
4.6 Diagrama de secuencia consultar base de datos.....	69
4.7 Diagrama de secuencia registrar eventos.....	70
4.8 Diagrama de secuencia estructurar un mensaje de texto.....	71
4.9 Diagrama de secuencia enviar mensajes.....	72
4.10 Diagrama de estado recibir mensajes.....	72
4.11 Diagrama de estado conexión con el teléfono celular.....	73
4.12 Diagrama de estado descargar mensajes de bandeja de entrada.....	73
4.13 Diagrama de estado analizar el mensaje de texto.....	74
4.14 Diagrama de estado consultar base de datos.....	74
4.15 Diagrama de estado registrar eventos.....	75
4.16 Diagrama de estado estructurar mensaje de texto.....	75
4.17 Diagrama de estado enviar mensaje.....	76
4.18 Diagrama de clases.....	77

RESUMEN

El presente trabajo trata sobre la creación de una aplicación que permita la consulta de las calificaciones de los alumnos de la Escuela Politécnica del Ejército sede Latacunga mediante mensajes de texto de un teléfono celular, para lo cual se investigó información relevante sobre la tecnología celular, de igual forma sobre los mensajes de texto y varios factores que inciden en esta tecnología.

Otra de las áreas que abarca el siguiente trabajo es la conexión del teléfono celular a un computador, para lo cual se tomó en cuenta aspectos como los puertos de comunicaciones, las formas a través de las cuales se puede conectar un teléfono celular al computador y el funcionamiento del teléfono como un modem.

Para la creación de la aplicación se empezó por analizar la herramienta de desarrollo a utilizar, el sistema operativo del computador empleado en la creación de la misma, las características del teléfono celular y del computador antes mencionado.

En la fase de codificación se partió de un análisis de los requisitos de la aplicación, posteriormente se realizó un modelamiento mediante diagramas UML, para finalmente llegar a la codificación de la aplicación y su puesta en funcionamiento.

CAPÍTULO I

LA TECNOLOGÍA CELULAR

1.1 INTRODUCCIÓN

Desde el principio de las telecomunicaciones dos han sido las principales opciones para llevar a cabo una comunicación: con o sin hilos, o lo que es lo mismo, por cable o por el aire. En realidad ambas pueden participar en un mismo proceso comunicativo.

En las comunicaciones móviles, en las que emisor o receptor están en movimiento, la movilidad de los extremos de la comunicación excluye casi por completo la utilización de cables para alcanzar dichos extremos. Por tanto utiliza básicamente la comunicación vía radio, esta se convierte en una de las mayores ventajas de la comunicación por la movilidad de los extremos de la conexión. Otras bondades de las redes inalámbricas son el ancho de banda que proporcionan, el rápido despliegue que conllevan al no tener que llevar a cabo obra civil.

La tecnología de radiotelefonía estaba limitada por dos factores muy importantes:

- En primer lugar solo hay un número limitado de canales en la RF (gama de frecuencias), la banda normal de telefonía podía dar servicio hasta 2000 canales.
- En segundo lugar la radiotelefonía era un servicio centralizado, uno se suscribía a un servicio local y solo se recibía servicio de ese proveedor, el alcance estaba limitado por las instalaciones del radio proveedor.

“Los servicios de comunicaciones móviles más extendidos son la telefonía móvil terrestre, la comunicación móvil por satélite, las redes móviles privadas, la radio mensajería, la radiolocalización GPS, las comunicaciones inalámbricas y el acceso a Internet móvil”¹

¹<http://www.eie.fceia.unr.edu.ar/ftp/Tecnologias%20de%20banda%20angosta/introduccion%20telefon%EDa%20celular%202003.pdf>.

Las tecnologías inalámbricas han tenido mucho auge y desarrollo en estos últimos años. Una con gran desarrollo ha sido la telefonía celular.

Desde sus inicios a finales de los 70 ha revolucionado enormemente las actividades que se realiza diariamente; los teléfonos celulares se han convertido en una herramienta primordial para la gente común y de negocios, las hace sentir más seguras y productivas.

A pesar de que la telefonía celular fue concebida estrictamente para la transmisión de voz, la tecnología celular de hoy es capaz de brindar otros tipos de servicios, como datos, audio y video con algunas limitaciones. “Sin embargo, la telefonía inalámbrica del mañana hará posible aplicaciones que requieran un mayor consumo de ancho de banda.”²

1.2 HISTORIA Y EVOLUCIÓN

A efectos de mostrar la evolución de la telefonía móvil en la historia, se procederá a describir una breve reseña que muestra el avance de la misma:

1843 – Un talentoso químico de nombre Michael Faraday comenzó un profundo estudio sobre la posible conducción de electricidad del espacio, lo que ayudó en forma incalculable en el desarrollo de la telefonía celular.

1876 - El teléfono es inventado por Alexander Graham Bell.

1894 – Si bien la comunicación inalámbrica tiene sus raíces en la invención del radio por Nikolai Tesla en la década de 1880, formalmente fue presentado en 1894 por un joven italiano llamado Gugliermo Marconi.

1947 - Los científicos desarrollaron las ideas que permitían el uso de teléfonos móviles usando "células" que identifican un usuario en cualquier punto desde donde se efectuara la llamada.

1949 – En la época predecesora a los teléfonos celulares, la gente que realmente necesitaba comunicación móvil tenía que confiar en el uso de radio-teléfonos en sus autos. En el sistema radio-telefónico, existía sólo una antena central por cada ciudad, y unos pocos canales disponibles en la torre.

² <http://www.revistaciencias.com/publicaciones/EpZyEkkuVlwxnCdrUU.php>

1964 – Los sistemas de telefonía móvil operaban sólo en el modo manual; un operador del teléfono móvil especial manejaba cada llamada, los sistemas selectores de canales automáticos fueron colocados en servicio para los sistemas de telefonía móvil.

1971 - AT&T hizo una propuesta sobre la posibilidad técnica de proporcionar un espectro de frecuencia mayor. Se comenzaba a delinear el principio de la radio celular.

1973 – El Dr. Martin Cooper es considerado el inventor del primer teléfono portátil, siendo gerente general de sistemas de Motorola realizó una llamada a sus competidores de AT&T desde su teléfono celular, transformándose en la primera persona en hacerlo.

1977 – Los teléfonos celulares se hacen públicos, dando comienzo las pruebas en el mercado.

1979 – Los primeros sistemas comerciales aparecieron en Tokio, Japón por la compañía NTT.

1983 – Chicago, Washington D.C. y Baltimore son los escenarios de los primeros lanzamientos de sistemas comerciales de telefonía celular en Estados Unidos.

– La AMPS (Sistema Avanzado de Telefonía Móvil) es lanzada usando frecuencias de banda desde 800 MHz. hasta 900 MHz y de 30 Khz. de ancho de banda para cada canal como un sistema totalmente automatizado de servicio telefónico. Es el primer estándar en telefonía celular en el mundo.

1986 – En varios países se diseminó la telefonía celular como una alternativa a la telefonía convencional inalámbrica. Debido a esta gran aceptación, el servicio comenzó a saturarse rápidamente.

1987 – La Industria llega a los 1000 millones de dólares en ganancias.

1988 – Se crea un nuevo estándar, el TDMA Interim Standard 54.

– Motorola introduce el teléfono móvil DynaTAC, el primer radioteléfono puramente "móvil".

1996 – Bell Atlantic Mobile lanza la primera red comercial CDMA en los Estados Unidos.

1997 – Los usuarios de la industria inalámbrica —celular, PCS y ESMR— superan los 50 millones.

– Entra en uso la red digital e inalámbrica de voz y datos (2G)

13 de Octubre de 2003 – 20° Aniversario de las Comunicaciones Inalámbricas Comerciales.

1.2.1 GENERACIONES DE LA TELEFONÍA CELULAR

Las distintas necesidades y avances dieron lugar a generaciones tecnológicas bien diferenciadas que se comentan a continuación.

En dicha evolución se aprecia cómo se van cumpliendo las necesidades del mercado para tener acceso múltiple al canal de comunicación, así como la necesaria migración de los sistemas analógicos a sistemas digitales con el fin de permitir mayor volumen de usuarios y ofrecer los niveles de seguridad que se demandaban.

1.2.1.1 Generación Cero (0G)

0G representa a la telefonía móvil previa a la era celular. Estos teléfonos móviles eran usualmente colocados en autos o camiones, también habían modelos en portafolios.

1.2.1.2 Primera generación (1G)

La 1G de la telefonía móvil hizo su aparición en 1979, si bien proliferó durante los años 80. Introdujo los teléfonos "celulares", basados en las redes celulares con múltiples estaciones de base relativamente cercanas unas de otras y protocolos para el "traspaso" entre las celdas cuando el teléfono se movía de una celda a otra.

La transferencia analógica y estrictamente para voz son características identificatorias de la generación, con enlaces muy reducidos, la velocidad de conexión no era mayor a (2400 bauds). En cuanto a la transferencia entre celdas, era muy imprecisa ya que contaban con una baja capacidad (Basadas en FDMA, Frequency Division Multiple Access), lo que limitaba en forma notable la cantidad de usuarios que el servicio podía ofrecer en forma simultánea ya que los protocolos de asignación de canal estáticos padecen de ésta limitación.

La tecnología predominante de esta generación es AMPS (Advanced Mobile Phone System). La información con la voz era transmitida en forma de frecuencia modulada al proveedor del servicio. Un canal de control era usado en forma simultánea para habilitar el traspaso a otro canal de comunicación de serlo necesario.

1.2.1.3 Segunda generación (2G)

Se convierte a un sistema digital, la primera llamada digital entre teléfonos celulares fue realizada en Estados Unidos en 1990. En 1991 la primera red GSM fue instalada en Europa. La generación se caracterizó por circuitos digitales de datos conmutados por circuito y la introducción de la telefonía rápida y avanzada a las redes. Usó a su vez acceso múltiple de tiempo dividido (TDMA) para permitir que hasta ocho usuarios utilizaran los canales separados por 200MHz. Los sistemas básicos usaron frecuencias de banda de 900MHz, mientras otros de 1800 y 1900MHz. Nuevas bandas de 850MHz fueron agregadas en forma posterior. El rango de frecuencia utilizado por los sistemas 2G coincidió con algunas de las bandas utilizadas por los sistemas 1G (como a 900Hz en Europa), desplazándolos rápidamente.

1.2.1.4 Generación 2.5 G

Una vez que la segunda generación se estableció, las limitantes de algunos sistemas en lo referente al envío de información se hicieron evidentes. Si bien la tercera generación estaba en el horizonte, algunos servicios se hicieron necesarios previa a su llegada. El General Packet Radio Service (GPRS) desarrollado para el sistema GSM fue de los primeros en ser visto. Hasta este momento, todos los circuitos eran dedicados en forma exclusiva a cada usuario. Este enfoque es conocido como "Circuit Switched", donde por ejemplo un circuito es establecido para cada usuario del sistema; esto era ineficiente cuando un canal transfería información sólo en un pequeño porcentaje. Aún más cantidad de mejoras fueron realizadas a la tasa de transferencia de información al introducirse el sistema conocido como EDGE (Enhanced Data rates aplicado a GSM Evolution). Éste básicamente es el sistema GPRS con un nuevo esquema de

modulación de frecuencia. Mientras GPRS y EDGE se aplicaron a GSM, otras mejoras fueron orientadas al sistema CDMA, siendo el primer paso de CDMA a CDMA2000 1x.

1.2.1.5 Tercera generación (3G)

Como suele ser inevitable, hay variados estándares con distintos competidores que intentan que su tecnología sea la predominante. Sin embargo, en forma muy diferencial a los sistemas 2G, el significado de 3G fue estandarizado por el proceso IMT-2000; este proceso no estandarizó una tecnología sino una serie de requerimientos (2 Mbit/s de máxima tasa de transferencia en ambientes cerrados, y 384 kbit/s en ambientes abiertos, por ejemplo). Existen principalmente tres tecnologías 3G. Para Europa existe UMTS (Universal Mobile Telecommunication System) usando CDMA de banda ancha (W-CDMA). Este sistema provee transferencia de información de hasta 2Mbps.

Están a su vez las evoluciones de CDMA2000. La primera en ser lanzada fue CDMA2000 1xEV-DO, donde EV-DO viene de Evolution Data Only. La idea atrás de este sistema era que muchas de las aplicaciones sólo requirieran conexión de datos, como sería el caso si se usara el celular para conectar una PC a Internet en forma inalámbrica. En caso de requerir además comunicación por voz, un canal 1X estándar es requerido. Además de usar tecnología CDMA, EV-DO usa tecnología TDMA para proveer de la velocidad de transferencia necesaria y mantener la compatibilidad con CDMA y CDMA2000 1X.

La siguiente evolución de CDMA2000 fue CDMA2000 1xEV-DV. Esto fue una evolución del sistema 1X totalmente distinto a CDMA2000 1xEV-DO, ofreciendo servicios totales de voz y datos. Este sistema también es compatible con CDMA y CDMA2000 1X y es capaz de ofrecer tasas de transferencia de 3.1Mbps.

Un tercer sistema 3G fue desarrollado en China que usa TDD. Conocido como TD-SCDMA (Time Division Synchronous CDMA), usa un canal de 1.6MHz y fue pensado para que abarque el mercado Chino y de los países vecinos.

1.3 TECNOLOGÍAS DE ACCESO CELULAR

En la actualidad existen tres tecnologías comúnmente usadas para transmitir información en las redes:

- Acceso múltiple por división de frecuencia (FDMA, por sus siglas en inglés)
- Acceso múltiple por división de tiempo (TDMA, por sus siglas en inglés)
- Acceso múltiple por división de código (CDMA, por sus siglas en inglés)

Aunque estas tecnologías suenan complicadas, se puede tener una idea de cómo funcionan examinando cada palabra de los nombres.

La diferencia primordial está en el método de acceso, el cual varía entre:

- Frecuencia, utilizada en la tecnología FDMA.
- Tiempo, utilizado en la tecnología TDMA.
- Códigos únicos, que se proveen a cada llamada en la tecnología CDMA.

La primera parte de los nombres de las tres tecnologías (Acceso múltiple), significa que más de un usuario (múltiple) puede usar (accesar) cada celda.

A continuación está detallado, sin entrar en complicados detalles técnicos, cómo funciona cada una de las tres tecnologías comunes.

La **tecnología FDMA** separa el espectro en distintos canales de voz, al separar el ancho de banda en pedazos (frecuencias) uniformes. La tecnología FDMA es mayormente utilizada para la transmisión analógica. Esta tecnología no es recomendada para transmisiones digitales, aun cuando es capaz de llevar información digital.

La **tecnología TDMA** comprime las conversaciones (digitales), y las envía cada una utilizando la señal de radio por un tercio de tiempo solamente. La compresión de la señal de voz es posible debido a que la información digital puede ser reducida de tamaño por ser información binaria (unos y ceros). Debido a esta compresión, la tecnología TDMA tiene tres veces la capacidad de un sistema analógico que utilice el mismo número de canales.

La **tecnología CDMA** es muy diferente a la tecnología TDMA. La CDMA, después de digitalizar la información, la transmite a través de todo el ancho de banda disponible. Varias llamadas son sobrepuestas en el canal, y cada una tiene un código de secuencia único. Usando la tecnología CDMA es posible comprimir entre 8 y 10 llamadas digitales para que estas ocupen el mismo espacio que ocuparía una llamada en el sistema analógico.

1.4 ESTÁNDARES

Desde los inicios de la telefonía celular en Estados Unidos, la Federal Communications Commission (FCC) adoptó un estándar analógico, el Advanced Mobile Phone System (AMPS), para asegurar la interoperabilidad de los diferentes sistemas celulares de primera generación.

Sin embargo, cuando las empresas de Personal Communications Service (PCS) empezaron a desplegar sus redes de segunda generación utilizando tecnologías digitales, las cuales permitían mejor calidad de señal y uso más eficiente de espectro que sus rivales analógicos, la FCC no forzó la utilización de un estándar único digital. De esta forma, a fines de los 80 Europa requería a los operadores utilizar el estándar de división de tiempo Global System for Mobile (GSM) para llamadas de voz

1.4.1 AMPS

El Sistema Telefónico Móvil Avanzado o AMPS (del inglés *Advanced Mobile Phone System*) es un sistema de telefonía móvil de primera generación (1G, voz analógica) desarrollado por los laboratorios Bell. Se implementó por primera vez en 1982 en Estados Unidos; se llegó a implantar también en Inglaterra y en Japón, con los nombres TACS y MCS-L1 respectivamente.

AMPS y los sistemas telefónicos móviles del mismo tipo dividen el espacio geográfico en una red de celdas o simplemente celdas (en inglés *cells*, de ahí el nombre de telefonía celular), de tal forma que las celdas adyacentes nunca usen las mismas frecuencias, para evitar interferencias. La estación base de cada celda emite con una potencia relativamente pequeña. La potencia emitida es tanto más pequeña, cuanto más pequeñas sean las celdas. Un pequeño tamaño de celda

favorece también la reutilización de frecuencias y aumenta, con mucho, la capacidad del sistema.

Sin embargo, también requiere un mayor número de estaciones base y por tanto una mayor inversión. Para poder establecerse la comunicación entre usuarios que ocupan distintas celdas se interconectan todas las estaciones base a un MTSO (*Mobile Telephone Switching Office*), también llamado MSC (*Mobile Switching Center*). A partir de allí se establece una jerarquía como la del sistema telefónico ordinario.

El uso de sistemas celulares da algunos problemas, como los que se plantean si el usuario cambia de celda mientras está hablando; AMPS prevé esto y logra mantener la comunicación activa siempre y cuando haya canales disponibles en la celda en la que se entra. Dependiendo del modo en el que se haga, puede cortarse la comunicación unos 300 ms para reanudarse inmediatamente después o puede ser completamente inapreciable para el usuario.

1.4.2 GSM

Es un estándar mundial para teléfonos celulares llamado *Global System for Mobile communications* (Sistema Global para las comunicaciones móviles), formalmente conocida como *Group Special Mobile* (**GSM**, Grupo Especial Móvil). En el año 2001, el 70% de los usuarios de telefonía móvil en el mundo usaban GSM. Es un estándar abierto, no propietario y que se encuentra en desarrollo constante.

GSM emplea una combinación de TDMA y FDMA entre estaciones en un par de canales de radio de frecuencia duplex. TDMA se utiliza para información digital codificada, por lo que GSM es un sistema diseñado para utilizar señales digitales, así como también, canales de voz digitales, lo que permite un moderado nivel de seguridad. Existen cuatro versiones principales basadas en la banda: GSM-850, GSM-900, GSM-1800 y GSM-1900, diferenciándose cada una en la frecuencia de las bandas. Las implementaciones más veloces de GSM se denominan **GPRS** y **EDGE**, también denominadas generaciones intermedias, o 2.5G, que conducen a la tercera generación (3G), o **UMTS**.

1. GPRS (*General Packet Radio Service*)

Básicamente es una comunicación basada en paquetes de datos. En GSM, los intervalos de tiempo son asignados mediante una conexión conmutada, en tanto que en GPRS son asignados mediante un sistema basado en la necesidad a la conexión de paquetes, es decir que, si no es enviado ningún dato por el usuario, las frecuencias quedan libres para ser utilizadas por otros usuarios. Los teléfonos GPRS por lo general utilizan un puerto bluetooth para la transferencia de datos.

2. EDGE (*Enhanced Data Rates for Global Evolution*)

Es una actualización de GPRS, el cual embala hasta 69.2Kbps en ocho timeslots, considerada una tecnología de 2.75G, un poco más evolucionada que GPRS. GERAN (*GPRS/EDGE Radio Access Network*) es el nombre que se le da a los estándares para el acceso GPRS/EDGE.

3. UMTS (*Universal Mobile Telecommunications System*)

“Es el sistema de telecomunicaciones móviles de tercera generación, que se espera que alcance unos 2000 millones de usuarios para el año 2010.”³

El principal avance radica en la tecnología WCDMA (*Wide Code Division Multiple Access*), heredada de la tecnología militar, a diferencia de GSM y GPRS que utilizan una mezcla de FDMA y TDMA. La principal ventaja de WCDMA es que la señal se expande en frecuencia gracias a un código de ensanchado que únicamente es conocido por el emisor y el receptor.

1.5 FABRICANTES

“Nokia es actualmente el mayor fabricante en el mundo de teléfonos móviles, con una cuota de mercado global de aproximadamente 39,1% en el segundo trimestre del 2008. Continúan la lista con un 14,4% Samsung, 10,2% Motorola, 8% LG Mobile y 7,5% Sony Ericsson.”⁴

Los principales fabricantes son:

³ <http://www.monografias.com/trabajos34/telefonía-celular/telefonía-celular2.shtml>

⁴ http://es.wikipedia.org/wiki/Telefonía_móvil

1.5.1 NOKIA

Además de una de las principales empresas del sector de las telecomunicaciones con sede en Keilaniemi de Espoo (Finlandia), es una de las marcas más conocidas dentro y fuera de la Unión Europea.

Nokia también produce infraestructura para redes de telefonía móvil y otros equipamientos de telecomunicaciones para aplicaciones como la telefonía tradicional por voz, RDSI, acceso a internet por banda ancha, radio profesional móvil, voz sobre IP y una línea de receptores de satélite. Nokia provee equipo de comunicaciones móvil para todo mercado considerable y protocolo, incluyendo GSM, CDMA y W-CDMA.

1.5.2 SAMSUNG

Es una de las empresas más fuertes y reconocidas de Corea del Sur a nivel mundial, y líder mundial en diversas ramas de la industria electrónica.

A pesar de ser más conocida como una empresa electrónica, Samsung está envuelta también en la industria pesada, automotriz, servicios financieros, productos químicos, venta al público y entretenimiento.

1.5.3 MOTOROLA

Es una empresa estadounidense especializada en la electrónica y las telecomunicaciones, establecida en Schaumburg, Illinois, un suburbio de Chicago. Los productos más conocidos de la compañía incluyen microprocesadores, teléfonos móviles y sistemas de conexión a redes de telefonía móvil e inalámbricas como Wimax y Wifi. Motorola es muy conocida por sus walkie talkies y también participó en el lanzamiento de la constelación de satélites Iridium.

1.5.4 LG ELECTRONICS

Fue constituida en 1958 y se enfoca en productos electrónicos, de información y de comunicaciones. Con sede en Seúl, Corea y filiales en 39 países alrededor del mundo, LG Electronics está compuesta por 4 empresas principales: Mobile Communications, Digital Appliance, Digital Display y Digital Media.

El objetivo de LG es aumentar su cuota de mercado en Europa, fortaleciendo su presencia en la telefonía móvil de tercera generación y la calidad de su imagen de marca.

1.5.5 SONY ERICSSON MOBILE COMMUNICATIONS AB

Es una empresa de riesgo compartido formada por la empresa sueca Ericsson y la empresa japonesa Sony dedicada a la fabricación de teléfonos móviles y accesorios.

La compañía se fundó en 2001 entre Ericsson, una de las empresas líderes en telecomunicaciones y Sony una de las empresas líderes en electrónica de consumo. Ambas empresas poseen partes iguales en Sony Ericsson.

1.5.6 APPLE INC

Es una empresa estadounidense de tecnología informática fundada en 1976, tuvo un rol prominente en la revolución informática de los años 70 con su computadora Apple II. Sus inicios son poco menos que curiosos, y forman parte del folclore informático.

1.6 EMPRESAS EN ECUADOR

En Ecuador existen tres proveedores de servicio de tecnología celular:

1.6.1 MOVISTAR

Antes conocida como BellSouth Ecuador, operó con el nombre de Otecel hasta que Telefónica Móviles, subsidiaria de Telefónica de España adquirió la totalidad de su capital accionario en octubre del 2004. Actualmente opera con la marca comercial Movistar y es proveedor de servicios inalámbricos de comunicaciones incluyendo servicios de voz, características mejoradas de llamadas, roaming internacional, internet inalámbrico, servicios de datos, intranets inalámbricas y otros servicios corporativos.

“*Movistar* es la segunda mayor operadora de telefonía móvil del Ecuador con más de 2,6 millones (a marzo del 2008) de clientes, con 89 puntos de atención al cliente y con redes AMPS, TDMA, CDMA y GSM.”⁵

En 2001 Otecel lanza al mercado el servicio de Internet móvil, y mensajes escritos (SMS), ofreciéndolo por un período de prueba gratuito a los clientes de cualquier plan pospago con un terminal compatible. Opera desde 2005 la red GSM de 850 MHz, impulsando una gran campaña de usar los teléfonos GSM de clientes que para entonces usaban Porta e incitándoles a comprar la SIM CARD (comúnmente denominado *chip*) usando su mismo equipo.

Telefónica Móviles es la empresa que gestiona los activos de telefonía móvil del Grupo Telefónica en 15 países de tres continentes. Al cierre del tercer trimestre de 2005 atendía a más de 89 millones de clientes, lo que le hace ser la segunda mayor multinacional del sector en todo el mundo.

A mediados del 2007 empieza instalación de un cable submarino de fibra óptica para lograr una tarifa más baja del servicio de internet, logrando llegar hasta un 50% menos. Telefónica es una empresa totalmente privada. Cuenta con más de 1,5 millones de accionistas directos.

1.6.2 PORTA

Porta es el nombre comercial de la empresa de telefonía móvil Consorcio Ecuatoriano de Telecomunicaciones (Conecel S.A.) de Ecuador que forma parte de la multinacional mexicana América Móvil. Es la compañía con más clientes a nivel nacional tanto de prepago como pospago (contrato), y así, la que más ingresos genera. Es además proveedor de internet.

Las negociaciones para el ingreso en el mercado ecuatoriano empezaron a finales de la década de los años ochenta. Se firma finalmente en 1993 un contrato de concesión de servicios de telecomunicaciones entre el Estado Ecuatoriano y CONECEL S.A. por medio de la Superintendencia de Telecomunicaciones de Ecuador (SUPTEL), por un período de 15 años. Su nombre comercial sería PORTA, que lo mantiene hasta ahora.

⁵ http://es.wikipedia.org/wiki/Telefónica_Móviles_Ecuador

Su domicilio fiscal se ubicaba en aquel entonces en la ciudad de Quito, pasando después a Guayaquil.

1.6.3 ALEGRO PCS

Alegro PCS es una compañía de telefonía celular de Ecuador. Filial de Andinatel(hoy CNT). Opera los servicios de telefonía móvil celular, servicio de internet, servicios portadores y de valor agregado.

La compañía Telecomunicaciones Móviles del Ecuador, Telecsa S.A., fue creada por Andinatel y Pacifictel para prestar el servicio de telefonía móvil. Inicialmente su capital estaba 50% en manos de cada compañía.

Recibió una concesión de parte del Estado ecuatoriano el 3 de abril de 2003 y entró en operaciones en diciembre de ese año. Su operación mejoró la competencia en el, hasta entonces, cerrado mercado celular: bajaron las tarifas, aumentaron los abonados, se interconectaron las redes para mensajes escritos, etc.

En diciembre de 2003 Alegro PCS comercializó un servicio que en Ecuador se llama Servicio Móvil Avanzado (SMA), en inglés con otras siglas se denomina PCS (Personal Communication System) al usar en espectro de 1900 Mhz, que se entiende es superior a un servicio celular. Usa tecnología CDMA 1X (EV-DO) en 1900 Mhz.

Alegro PCS en el primer cuarto del 2005 tenía listo su servicio de Internet inalámbrico NIU Internet Total, el primero en el país, el cual ofrecía una velocidad de 70kbps promedio con los mismos terminales que los usados para los servicios de voz y SMS.

No obstante de que CDMA es una tecnología definitivamente más eficiente y sofisticada, el éxito comercial de GSM en la región ha llevado a Alegro (ya sin el aditivo PCS) a tener un arreglo con Movistar alquilándole las antenas y de esta manera prestar también servicios en GSM (diciembre de 2007) sin incurrir en los altos costos de montar una red paralela más aun cuando se estima que en próximos años las actuales redes GSM serían reemplazadas por redes de tercera generación.

1.7 MENSAJES DE TEXTO (SMS)

1.7.1 ASPECTOS IMPORTANTES

1.7.1.1 Definición

El servicio de mensajes cortos o SMS (*Short Message Service*) es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos (también conocidos como mensajes de texto) entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano. SMS fue diseñado originariamente como parte del estándar de telefonía móvil digital GSM, pero en la actualidad está disponible en una amplia variedad de redes, incluyendo las redes 3G.

1.7.1.2 Historia

La idea de añadir la mensajería de texto a los servicios de usuarios móviles era latente en muchas comunidades de servicios de comunicación móviles al principio de los años 1980. Los expertos de varias de aquellas comunidades contribuyeron en las discusiones sobre cuáles deberían ser los servicios GSM. En un principio el SMS se pensó como una manera de avisar al usuario, por ejemplo, de llamadas perdidas o mensajes en el buzón de voz, sin embargo pocos creyeron que el SMS sería usado como el medio para enviar mensajes de texto de un usuario móvil a otro.

En febrero de 1985 el SMS fue considerado en su mayoría GSM, como un servicio posible para el nuevo sistema digital celular.

Las discusiones sobre los servicios GSM entonces fueron concluidas en la recomendación GSM 02.03 "TeleServices supported by a GSM PLMN". Aquí dieron una descripción rudimentaria de los tres servicios:

1. Móvil de mensaje corto Terminado (SMS-MT): la capacidad de una red para transmitir un Mensaje Corto a un teléfono móvil. El mensaje puede ser enviado por un teléfono o por una aplicación de software.
2. Móvil de mensaje corto Originado (SMS-MO): la capacidad de una red para transmitir un Mensaje Corto enviado por un teléfono móvil. El mensaje puede ser enviado a un teléfono o a una aplicación de software.
3. Difusión de Célula de mensaje corta.

El primer mensaje comercial SMS fue enviado por la red GSM de Vodafone el 3 de diciembre de 1992 en el Reino Unido a través de un ordenador, el texto del mensaje era “la navidad alegre”.

“Comercialmente el SMS es una industria masiva que en el 2006 merece más de 80 mil millones de dólares a escala mundial. SMS tiene un precio medio global de 11 centavos y mantiene un margen de beneficio del 90 %.”⁶

1.7.1.3 Funcionamiento de los SMS

Esto tiene mucho que ver de cómo funcionan los móviles en sí. Incluso si no se está hablando por un teléfono móvil, este dispositivo está constantemente mandando y recibiendo información. Está “hablando” con su antena o torre de telefonía sobre una conexión llamada canal de control. Esta antena o torre puede estar encima de un bloque de viviendas o en algún edificio cercano. La razón de esto es para que el sistema de móviles sepa en qué celda está el teléfono y para ir cambiando de celda según se vaya moviendo. Cada cierto tiempo, el teléfono y la antena intercambiarán un paquete de datos y se harán saber que todo está perfecto.

El teléfono también usa el canal de control para la preparación de la llamada. Cuando alguien intenta llamar, la antena o torre envía un mensaje sobre el canal de control y le dice al teléfono que haga sonar el timbre que se tiene puesto. La torre también le da al teléfono un par de canales de frecuencia de voz para ser utilizados por la llamada.

El canal de control provee también el camino para los mensajes SMS. Cuando un usuario envía un SMS, el mensaje fluye a través del SMSC (centro de servicios de mensajes cortos que gestiona los intercambios de SMS), y después a la torre, y la torre entonces envía el mensaje al teléfono como un pequeño paquete de datos sobre el canal de control. De la misma manera, cuando se envía un mensaje, el teléfono lo envía a la torre sobre el canal de control, y desde la torre al SMSC y desde ahí al destino.

⁶ http://es.wikipedia.org/wiki/Historia_del_SMS

Las funciones del SMSC son:

- Recibir y almacenar los mensajes cortos enviados por los usuarios o por otras fuentes (avisos del operador, buzón de voz, sistemas de publicidad, alertas de correo electrónico, etc.) hasta que puedan ser enviados;
- Verificar los permisos del usuario para enviar mensajes.
- Verificar si el usuario al que se envía el mensaje está operativo o no; si está operativo, el mensaje se envía, y si no se almacena temporalmente en el SMSC;
- Verificar periódicamente el estado de los usuarios que tienen mensajes pendientes.

1.7.1.4 Ventajas del SMS

Los mensajes SMS tienen varias ventajas. Es más discreto que una conversación de voz, haciéndolo ideal como forma de comunicación cuando no quieres que te oigan. Normalmente lleva menos tiempo enviar un mensaje de este tipo que realizar una llamada o enviar un email. Un SMS no requiere que estés en tu ordenador para hacer el envío.

“Los SMS también son un buen sistema para que gente con problemas de audición se comuniquen.”⁷

Los SMS son un servicio de almacenamiento y envío, que significa que cuando se envía un mensaje de texto a un usuario, dicho mensaje no va directamente a su teléfono móvil. La ventaja de este método es que el teléfono móvil de un usuario no tiene por qué estar encendido o con cobertura para que le envíen mensajes. El mensaje se almacena en SMSC los días que sean necesarios, hasta que el teléfono se encienda o tenga cobertura, donde el mensaje es recibido. Este mensaje permanecerá en la tarjeta SIM del receptor hasta que lo borre.

Aparte de hacer envío de persona a persona, los SMS se pueden usar para enviar mensajes a un número grande de gente al mismo tiempo, ya sea por medio de una lista de contactos o todos los usuarios en un área particular.

⁷ <http://www.ordenadores-y-portatiles.com/informacion-sms.html>

1.7.2 COMPONENTES

Cuando un usuario envía un SMS, o lo recibe, se incluyen con su payload (carga útil o cuerpo del mensaje) al menos los siguientes parámetros:

- Fecha de envío (también llamada *timestamp*);
Timestamp es una secuencia de caracteres que denotan la hora y fecha (o alguna de ellas) en la cual ocurrió determinado evento. Esta información es comúnmente presentada en un formato consistente, lo que permite la fácil comparación entre dos diferentes registros y seguimiento de progresos a través del tiempo.
- Validez del mensaje, desde una hora hasta una semana;
- Número de teléfono del remitente y del destinatario;
- Número del SMSC (centro de mensajes cortos) que ha originado el mensaje;

De este modo se asegura el correcto procesamiento del mensaje en el SMSC y a lo largo de toda la cadena.

1.7.3 FORMATOS

Existen dos formatos para enviar SMS, algunos dispositivos no soportan los dos:

1.7.3.1 SMS en formato texto

Este formato es el más sencillo ya que no está de ningún modo manipulado el mensaje y se envía tal cual se escribe (originalmente de 160 caracteres o menos, incluyendo los espacios), el formato que soporta el texto es ASCII.

1.7.3.2 SMS en formato PDU

El modo PDU trata el SMS como una cadena de caracteres en octetos hexadecimales o semioctetos decimales, de cuya codificación resulta el SMS en modo texto. La ventaja de modo PDU respecto al modo texto es que en modo texto la aplicación queda limitada a la opción de codificación que se haya preestablecido, en modo PDU se puede implementar cualquier codificación.

La cadena PDU no solo contiene el mensaje, sino que lleva información del centro de servicio SMS, hora de llegada, tipo de mensaje, información sobre el que envía el SMS, vigencia del SMS, nº de caracteres del SMS, tipo de llamada (nacional ó internacional), tipo de alfabeto usado.

1.7.3.2.1 Estructura de los SMS en formato PD.

SMS Enviados



Tabla 1.1: Estructura de un SMS enviado.

DCS: Dirección Centro de Servicio

TIPO PDU: Tipo Protocolo de la Unidad de Datos



Tabla 1.2: Elementos TIPO PDU SMS enviado

PC: Path contestación (0 No, 1 Sí)

CD: Cabecera datos (0 Sin, 1 Con)

PRE: Petición reporte de estado (0 No, 1 Sí)

PV: Campo Periodo Vigencia presente

RD: Permite que el centro de SMS acepte un SMS-SUBMIT para un mensaje que todavía está en el centro. (0 Sí, 1 No)

TIPO: Mensaje de envío

DD: Dirección destino. Se rellenará igual que el campo Dirección centro de servicio, poniendo el nº de teléfono del destinatario del SMS.

PID: Protocolo Identificación

NR: Número de referencia

COD: Codificación trama de datos. Indica el alfabeto con el que se codifica la trama (alfabeto por defecto= codificación a 7 bits o codificación a 8 bits) e indica tipo de SMS.

PV: Período de Vigencia del SMS

LD: Longitud de la cadena de datos

Ejemplo: si el mensaje está formado por la siguiente cadena C8 27 33 08
El campo LD se rellenará con 04 (ya que hay 4 octetos= 2 caracteres hexadecimal)

DATOS: Para explicar la codificación a 7 BITS se usará un ejemplo. Codificación de la palabra HOLA:

	H	O	L	A
Hex	48	4F	4C	41
Bin	1001000	1001111	1001100	1000001

Tabla 1.3: Codificación a 7 bits

Para transformar a octetos se toma el número de caracteres de la siguiente letra que nos falten para llegar a 8, cuando se hayan tomado caracteres de una letra para la anterior, esta se queda sin esos caracteres y los debe tomar de la siguiente letra

1 1001000	00 100111 1	001 10011 00	1000 001
C8	27	33	08

Tabla 1.4: Transformación a octetos

Ejemplo Envío SMS en formato PDU

Envío de un SMS con la palabra ALARMA

	A	L	A	R	M	A
Hex	41	4C	41	52	4D	41
Bin	1000001	100110 0	10000 01	1010 010	100 1101	10 00001
	0 1000001	01 100110 0	010 10000 01	1101 1010 010	00001 100 1101	10 00001
	41	66	50	DA	0C	02

Tabla 1.5: Ejemplo de envío de SMS en formato PDU

SMS Recibidos

DCS	TIPO	DO	PID	NR	COD	TR	LD	DATOS
	PDU							

Tabla 1.6: Estructura de un SMS recibido

DCS: Dirección Centro de Servicio. Será el nº centro de servicio de envío al que pertenezca el teléfono

TIPO PDU: Tipo Protocolo de la Unidad de Datos

PC	CD	RE	sin uso	RD	TIPO
----	----	----	---------	----	------

Tabla 1.7: Elementos TIPO PDU SMS recibido

PC: Path contestación (0 No, 1 Sí)

CD: Cabecera datos (0 Sin, 1 Con)

RE: Petición reporte de estado (0 No, 1 Sí). Este campo lo rellena el centro de servicio.

RD: Permite que el centro de SMS acepte un SMS-SUBMIT para un mensaje que todavía está en el centro. (0 Sí, 1 No)

TIPO: Tipo SMS recibido

DO: Dirección Origen. Este campo contiene el nº del que envía el SMS

PID: Protocolo Identificación

COD: Codificación trama de datos. Indica qué tipo de codificación tienen los datos

TR: Tiempo Recepción. Indica cuando se ha recibido el SMS. Invierte el orden por pares de caracteres.

Año	Mes	Día	Hora	Minutos	Segundos	Zona Horaria
-----	-----	-----	------	---------	----------	--------------

Tabla 1.8: Estructura del tiempo de recepción SMS recibido

Ejemplo. Supóngase la fecha/hora: 14/03/02 a las 20:17:52

La codificación sería 20 30 41 02 71 25 00

LD: Longitud Datos

DATOS: Datos codificados (con el alfabeto indicado en el campo codificación de datos)

Ejemplo recepción de SMS

Recibimos el SMS con el texto “ALARMA” el: 28/05/02 a las 15:24:59 zona 00
+CMGL: 12,1,,25

07914356060018F2040B914356554985F200002050825142950006416650D
A0C02

Donde:

07 = nº octetos que forman el campo dirección centro servicio

91 = nº con prefijo internacional

4356060018F2 = nº centro servicio 34656000812

04 = tipo PDU (SMS recibido)

0B = número de semioctetos (caracteres) del teléfono de envío

91 = nº sin prefijo internacional

4356554985F2 = nº teléfono de envío +34655594582

00 = protocolo identificación

00 = codificación trama de datos a 7 bits. Alfabeto por defecto

20508251429500 Cuando llegó el SMS: 28/05/02, 15:24:59+00

06 = nº de octetos que forman los datos del mensaje

416650DA0C02 = datos “ALARMA” codificado a 7 bits

1.7.4 ESTÁNDARES

SMS fue diseñado originariamente como parte del estándar de telefonía móvil digital GSM, pero en la actualidad está disponible en una amplia variedad de redes, incluyendo las redes 3G.

1.7.4.1 GSM

Un mensaje SMS es una cadena alfanumérica de hasta 160 caracteres de 7 bits, y cuyo encapsulado incluye una serie de parámetros. En principio, se emplean para enviar y recibir mensajes de texto normal, pero existen extensiones del protocolo básico que permiten incluir otros tipos de contenido, dar formato a los mensajes o encadenar varios mensajes de texto para permitir mayor longitud.

En GSM existen varios tipos de mensajes de texto: mensajes de texto "puros", mensajes de configuración (que contienen los parámetros de conexión para otros servicios, como WAP o MMS), notificaciones de mensajes MMS

En otros estándares de telefonía móvil (como CDMA2000 o UMTS) el proceso de los mensajes se realiza de otra forma, pero el funcionamiento es transparente de cara al usuario.

1.7.4.2 CDMA2000

CDMA2000 es una familia de estándares de telecomunicaciones móviles de tercera generación (3G) que utilizan CDMA, un esquema de acceso múltiple para redes digitales, para enviar voz, datos, y señalización (como un número telefónico marcado) entre teléfonos celulares y estaciones base.

CDMA (*code division multiple access* ó acceso múltiple por división de código) es una estrategia de multiplexado digital que transmite flujos de bits. Básicamente, CDMA permite que múltiples terminales compartan el mismo canal de frecuencia, identificándose el "canal" de cada usuario mediante (secuencias PN).

“CDMA2000 ha tenido relativamente un largo historial técnico y aún sigue siendo compatible con los antiguos estándares en telefonía CDMA (como cdmaOne) primero desarrollado por Qualcomm, una compañía comercial, y propietario de varias patentes internacionales sobre la tecnología.”⁸

1.7.4.3 UMTS

Sistema Universal de Telecomunicaciones móviles (Universal Mobile Telecommunications System - UMTS) es una de las tecnologías usadas por los móviles de tercera generación (3G, también llamado W-CDMA), sucesora de GSM.

Aunque inicialmente esté pensada para su uso en teléfonos móviles, la red UMTS no está limitada a estos dispositivos, pudiendo ser utilizada por otros.

“Sus tres grandes características son las capacidades multimedia, una velocidad de acceso a Internet elevada, la cual además le permite transmitir audio y video

⁸ <http://es.wikipedia.org/wiki/CDMA2000>

en tiempo real; y una transmisión de voz con calidad equiparable a la de las redes fijas. Pero dispone de una variedad de servicios muy extensa.”⁹

UMTS permite introducir muchos más usuarios a la red global del sistema, y además permite incrementar la velocidad a 2 Mbps por usuario móvil.

UMTS ofrece los siguientes servicios:

- Facilidad de uso y bajos costes: UMTS proporcionará servicios de uso fácil y adaptable para abordar las necesidades y preferencias de los usuarios, amplia gama de terminales para realizar un fácil acceso a los distintos servicios y bajo coste de los servicios para asegurar un mercado masivo.
- Nuevos y mejorados servicios: Los servicios de voz mantendrán una posición dominante durante varios años. Los usuarios exigirán a UMTS servicios de voz de alta calidad junto con servicios de datos e información. Un ejemplo de esto es la posibilidad de conectarse a Internet desde el terminal móvil o desde el ordenador conectado a un terminal móvil con UMTS.
- Acceso rápido: La principal ventaja de UMTS sobre la segunda generación móvil (2G), es la capacidad de soportar altas velocidades de transmisión de datos de hasta 144 kbit/s sobre vehículos a gran velocidad, 384 kbit/s en espacios abiertos y 7.2 Mbit/s con baja movilidad (interior de edificios). Esta capacidad sumada al soporte inherente del protocolo de Internet (IP), se combinan poderosamente para prestar servicios multimedia interactivos y nuevas aplicaciones de banda ancha.

⁹ http://es.wikipedia.org/wiki/Universal_Mobile_Telecommunications_System

1.7.5 COSTOS DEL SERVICIO

Movistar¹⁰

Cantidad de mensajes	Precio sin (IVA)	Precio Final
30	\$1.00	\$1.12
50	\$1.50	\$1.68
70	\$2.00	\$2.24
150	\$3.50	\$3.92
250	\$5.50	\$6.16
320	\$6.50	\$7.28
400	\$7.69	\$8.61
2500	\$12.00	\$13.44

Tabla 1.9: Costos SMS Movistar

- Tarifa SMS adicional movistar a movistar \$0.06 sin impuestos (IVA). Precio final \$0.067.
- Tarifa SMS movistar a otras operadoras movistar \$0.06 sin impuestos (IVA). Precio final \$0.067.
- Tarifa SMS internacional \$0.10 sin impuestos (IVA). Precio final \$0.112.

Porta¹¹

Cantidad de mensajes	Precio sin (IVA)	Precio Final
30	\$1.00	\$1.12
50	\$1.50	\$1.68
90	\$2.00	\$2.24
160	\$3.25	\$3.64
240	\$5.00	\$5.60
2800	\$11.99	\$13.43

Tabla 1.10: Costos SMS Porta

¹⁰ Tomado de la página web oficial de movistar en Ecuador <http://www.movistar.com.ec/> en marzo 2010

¹¹ Tomado de la página web oficial de Porta en Ecuador <http://www.porta.net/> en marzo 2010

- Los mensajes de porta a otra operadora tiene un costo de \$0,06 + IVA por SMS
- Los mensajes internacionales tienen un costo de **\$0.10 + Impuestos** y se cobrará por eventos.

Alegro¹²

Cantidad de mensajes	Precio sin (IVA)	Precio Final
30	\$0.75	\$0.84
120	\$1.75	\$1.96
220	\$2.75	\$3.08
450	\$4.75	\$5.32
Ilimitado	\$7.75	\$8.68

Tabla 1.11: Costos SMS Alegro

- Sms alegro – alegro \$0,05 IVA. Precio Final: \$0.056
- Sms alegro – otras operadoras \$0,06 IVA por mensaje enviado. Precio Final: \$0,067
- Sms internacional \$0,10 IVA por mensaje enviado. Precio Final: \$0.112. Interconexión incluida. Los mensajes entrantes no tienen costo para el abonado.

¹² Tomado de la página web oficial de Alegro en Ecuador <http://www.alegro.com.ec/> en marzo 2010

CAPÍTULO II

CONECTIVIDAD ENTRE EL PC Y EL TELÉFONO CELULAR

2.1 PUERTOS DE COMUNICACIONES

Hoy en día las computadoras han evolucionado desde que se inventó la primera y con ellas han avanzado los dispositivos de almacenamiento. Debido al avance tecnológico se crearon puertos que sirven para recibir y enviar datos de la computadora a periféricos que estén conectados a ella, estos se llaman puertos de comunicación y actualmente se conoce una gran gama de ellos.

“Los puertos de comunicación son herramientas que permiten manejar e intercambiar datos entre un computador (generalmente están integrados en las tarjetas madres) y sus diferentes periféricos o entre dos computadores.”¹³

Entre los principales puertos de comunicación se tiene:

2.1.1 PUERTOS PS/2

2.1.1.1 Definición

Estos puertos son en esencia puertos serie que se utilizan para conectar pequeños periféricos a la PC. Su nombre viene dado por las computadoras de modelo IBM Personal System/2 de IBM, donde fueron utilizados por primera vez.

2.1.1.2 Características

Este es un puerto serial, con conectores de tipo Mini DIN, el cual consta por lo general de 6 pines o conectores, la placa base tiene el conector hembra. En las placas de hoy en día se pueden distinguir el teclado del Mouse por sus colores, siendo el teclado (por lo general) el de color violeta y el Mouse el de color verde.

¹³ <http://www.monografias.com/trabajos33/puertos-de-comunicacion/puertos-de-comunicacion.shtml?monosearch>

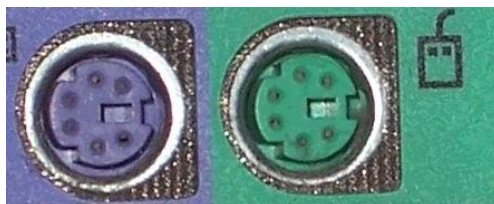


Figura 2.1: Puertos PS/2

2.1.1.3 Forma

Existen 2 conectores diferentes para estos puertos; el primero es un DIN de 5 pines (conocido comúnmente como AT) y el segundo es un conector Mini DIN de 6 pines (normalmente llamado PS/2). Estos dos conectores son electrónicamente iguales, lo único que cambia es su apariencia interna.

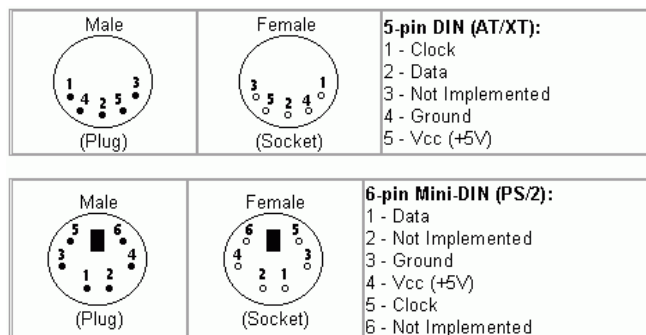


Figura 2.2: Forma de puertos PS/2

2.1.1.4 Ubicación en el sistema informático

Estos puertos son utilizados principalmente por teclados y mouse.

2.1.2 PUERTOS USB (UNIVERSAL SERIAL BUS)

2.1.2.1 Definición

El *Universal Serial Bus* (bus universal en serie) o Conductor Universal en Serie, abreviado comúnmente USB, es un puerto que sirve para conectar periféricos a una computadora. Fue creado en 1996.

Es una arquitectura de bus desarrollada por las industrias de computadoras y telecomunicaciones que permite instalar periféricos sin tener que abrir la máquina para instalarle hardware, es decir, que basta con conectar dicho periférico al computador.

El estándar incluye la transmisión de energía eléctrica al dispositivo conectado. Algunos dispositivos requieren una potencia mínima, así que se pueden conectar varios sin necesitar fuentes de alimentación extra. Los concentradores con fuente de alimentación pueden proporcionarle corriente eléctrica a otros dispositivos sin quitarle corriente al resto de la conexión (dentro de ciertos límites).

2.1.2.2 Características

- Una central USB le permite adjuntar dispositivos periféricos rápidamente, sin necesidad de reiniciar la computadora ni de volver a configurar el sistema.
- El USB trabaja como interfaz para la transmisión de datos y distribución de energía.
- Los periféricos para puertos USB son reconocidos automáticamente por el computador (y se configuran casi automáticamente) lo cual evita dolores de cabeza al instalar un nuevo dispositivo en el PC.

Los dispositivos USB se clasifican en cuatro tipos según su velocidad de transferencia de datos:

- Baja velocidad (1.0): Tasa de transferencia de hasta 1'5 Mbps (192 KB/s).
- Velocidad completa (1.1): Tasa de transferencia de hasta 12 Mbps (1'5 MB/s). Ésta fue la más rápida antes de la especificación USB 2.0.
- Alta velocidad (2.0): Tasa de transferencia de hasta 480 Mbps (60 MB/s).
- Super alta velocidad (3.0): Actualmente en fase experimental y con tasa de transferencia de hasta 4.8 Gbps (600 MB/s).

2.1.2.3 Forma

Existe un solo tipo de cable USB (A-B) con conectores distintos en cada extremo, de manera que es imposible conectarlo erróneamente. Consta de 4 hilos, transmite a 12 Mbps y es "Plug and Play", que distribuye 5v para alimentación y transmisión de datos.



Figura 2.3: Conectores USB

2.1.2.4 Ubicación en el sistema informático

El USB es la tecnología preferida para la mayoría de los teclados, mouse y otros dispositivos de entrada de información de banda estrecha. El USB también está muy extendido en cámaras fotográficas digitales, impresoras, escáneres, módems, joysticks y similares.

2.1.3 PUERTOS SERIALES (COM)

2.1.3.1 Definición

Son adaptadores que se utilizan para enviar y recibir información de BIT en BIT fuera del computador a través de un único cable y de un determinado software de comunicación. Un ordenador o computadora en serie es la que posee una unidad aritmética sencilla en la cual la suma en serie es un cálculo dígito a dígito

2.1.3.2 Características

- Los puertos seriales se identifican típicamente dentro del ambiente de funcionamiento como puertos COM (comunicaciones). Por ejemplo, un mouse pudo ser conectado con COM1 y un módem a COM2.
- Los voltajes enviados por los pines pueden ser en 2 estados, encendido o apagado. Encendido (valor binario de 1) significa que el pin está transmitiendo una señal entre -3 y -25 voltios, mientras que apagado (valor binario de 0) quiere decir que está transmitiendo una señal entre +3 y +25 voltios.

Existen tres tipos de comunicaciones seriales:

- Simplex.- En este caso el transmisor y el receptor están perfectamente definidos y la comunicación es unidireccional.
- Duplex, half duplex o semi-duplex.- En este caso ambos extremos del sistema de comunicación cumplen funciones de transmisor y receptor y los datos se desplazan en ambos sentidos pero no simultáneamente.
- Full Duplex .-El sistema es similar al duplex, pero los datos se desplazan en ambos sentidos simultáneamente. Para ello ambos transmisores

poseen diferentes frecuencias de transmisión o dos caminos de comunicación separados,

2.1.3.3 Forma

Estos conectores son de tipo macho y los hay de 2 tamaños, uno estrecho, de 9 pines agrupados en dos hileras con una longitud aproximada de 17mm y otro ancho de 25 pines, con una longitud de unos 38mm, internamente son iguales (9 pines) y realizan las mismas funciones.

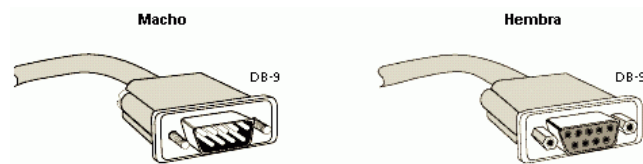


Figura 2.4: Conectores DB-9 macho y hembra

2.1.3.4 Ubicación en el sistema informático

Estos puertos se utilizan para conectar el mouse y el módem. Normalmente el mouse se conecta a un puerto COM de 9 pines (comúnmente COM1) y el módem se conecta a un puerto de 25 pines (comúnmente COM2).

2.1.4 PUERTOS RJ-45

2.1.4.1 Definición

Es una interfaz física utilizada comúnmente en las redes de computadoras, sus siglas corresponden a "Registered Jack" o "Clavija Registrada", que a su vez es parte del código de regulaciones de Estados Unidos.

2.1.4.2 Características

Para que todos los cables funcionen en cualquier red, se sigue un estándar a la hora de hacer las conexiones.

- Este conector se utiliza en la mayoría de las tarjetas de ethernet (tarjetas de red) y va en los extremos de un cable UTP nivel 5.

2.1.4.3 Forma

Posee ocho pines o conexiones eléctricas, que normalmente se usan como extremos de cables de par trenzado.

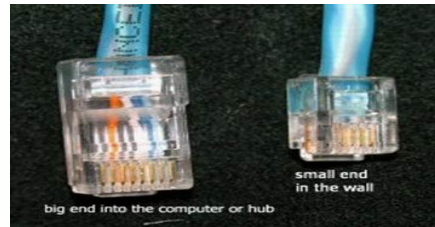


Figura 2.5: Conectores RJ-45

2.1.4.4 Ubicación en el sistema informático

Se conecta a la tarjeta de red. Puede tener el formato RJ45 (parecido al de un conector de teléfono).

2.2 CONEXIÓN ENTRE EL CELULAR Y EL PC

En la actualidad existen tres formas de conectar un teléfono celular al PC:

2.2.1 CABLE USB

El cable USB es el medio más utilizado para conectar un teléfono celular al PC, por su capacidad de transferir datos a altas velocidades, por ser un puerto que posee casi todo computador en la actualidad y por que la gran mayoría de teléfonos celulares se conecta a este puerto de comunicaciones.

Otra de las ventajas de usar este puerto de comunicaciones es que maneja voltajes de hasta 5v, por lo que algunos fabricantes de teléfonos han aprovechado esta característica, haciendo que mientras el celular se encuentre conectado al computador a través del puerto USB la batería del mismo se cargue.

El principal inconveniente de usar este tipo de medio es que la mayoría de teléfonos celulares poseen en uno de los extremos del cable un conector de diseño propio de cada marca y para cada modelo de celular, en el otro extremo un conector USB estándar.

2.2.2 BLUETOOTH

Esta tecnología permite la comunicación sin la necesidad de cables y con un bajo consumo de energía.

Su uso en la actualidad se encuentra en crecimiento por las ventajas que ofrece pero sin llegar a la gran aceptación que tiene el uso del cable USB debido a que la tecnología bluetooth no se encuentra en todos los teléfonos celulares y que pocos PCs cuentan con ésta incorporada en los mismos.

2.2.2.1 Descripción

Bluetooth intenta unir diferentes tecnologías como la de las computadoras, los teléfonos móviles y el resto de periféricos.

Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPANs) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia segura y globalmente libre (2,4 GHz.). Los principales objetivos que se pretenden conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre nuestros equipos personales.

Los dispositivos que con mayor intensidad utilizan esta tecnología son los de los sectores de las telecomunicaciones y la informática personal, como PDAs, teléfonos móviles, computadoras portátiles, ordenadores personales, impresoras y cámaras digitales.

Bluetooth se denomina al protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, con una cobertura baja y basados en transceptores de bajo coste.

Gracias a este protocolo, los dispositivos que lo implementan pueden comunicarse entre ellos cuando se encuentran dentro de su alcance. Las comunicaciones se realizan por radiofrecuencia de forma que los dispositivos no tienen por qué estar alineados, pueden incluso estar en habitaciones separadas si la potencia de transmisión lo permite.

2.2.2.2 Especificaciones

La clasificación de los dispositivos Bluetooth como "Clase 1", "Clase 2" o "Clase 3" es únicamente una referencia de la potencia de transmisión del dispositivo, siendo totalmente compatibles los dispositivos de una clase con los de la otra.

Clase	Potencia máxima permitida (mW)	Potencia máxima permitida (dBm)	Rango (aproximado)
Clase 1	100 mW	20 dBm	~100 metros
Clase 2	2.5 mW	4 dBm	~10 metros
Clase 3	1 mW	0 dBm	~1 metro

Tabla 2.1: Clases de Bluetooth

En cuanto al ancho de banda:

Versión	Ancho de banda
Versión 1.2	1 Mbit/s
Versión 2.0 + EDR	3 Mbit/s
UWB Bluetooth (propuesto)	53 - 480 Mbit/s

Tabla 2.2: Versiones de Bluetooth

Las versiones existentes son:

- Bluetooth v.1.1
- Bluetooth v.1.2
- Bluetooth v.2.0
- Bluetooth v.2.1
- Bluetooth v.2.2

La versión 1.2, a diferencia de la 1.1, provee una solución inalámbrica complementaria para co-existir Bluetooth y Wi-Fi en el espectro de los 2.4 GHz, sin interferencia entre ellos.

La versión 1.2 usa la técnica "Adaptive Frequency Hopping (AFH)", que ejecuta una transmisión más eficiente y un cifrado más seguro.

La versión 2.0, creada para ser una especificación separada, principalmente incorpora la técnica "Enhanced Data Rate" (EDR) que le permite mejorar las velocidades de transmisión en hasta 3Mbps a la vez que intenta solucionar algunos errores de la especificación 1.2.

La versión 2.1, simplifica los pasos para crear la conexión entre dispositivos, además el consumo de potencia es 5 veces menor.

La versión 2.2 aumenta considerablemente la velocidad de transferencia. La idea es que el nuevo Bluetooth trabaje con WiFi, de tal manera que sea posible lograr mayor velocidad.

El hardware que compone el dispositivo Bluetooth está compuesto por dos partes:

- Un dispositivo de radio, encargado de modular y transmitir la señal.
- Un controlador digital, compuesto por una CPU, por un procesador de señales digitales (DSP - Digital Signal Processor) llamado Link Controller (o controlador de Enlace) y de los interfaces con el dispositivo anfitrión.

2.2.3 ENLACE INFRARROJO

Esta es otra tecnología inalámbrica, en la actualidad cada vez menos dispositivos usan esta tecnología debido entre otras razones a que necesita línea directa de vista entre los dispositivos que necesiten conectarse, cabe mencionar que no todos los teléfonos celulares cuentan con esta tecnología y de igual forma los PCs necesitan de dispositivos adicionales para tener un enlace infrarrojo.

2.2.3.1 Descripción

Los enlaces infrarrojos se encuentran limitados por el espacio y los obstáculos. El hecho de que la longitud de onda de los rayos infrarrojos sea tan pequeña (850-

900 nm), hace que no pueda propagarse de la misma forma en que lo hacen las señales de radio.

Por otro lado, las transmisiones infrarrojas presentan la ventaja, frente a las de radio, de no transmitir a frecuencias bajas, donde el espectro está más limitado, no teniendo que restringir, por tanto, su ancho de banda a las frecuencias libres.

2.2.3.2 Características

A la hora de transmitir, las estaciones infrarrojas pueden usar tres tipos de métodos para ello: punto a punto, casi-difuso y difuso.

En el modo *punto a punto*, el tipo de emisión por parte del transmisor se hace de forma direccional. Por ello, las estaciones deben verse directamente, para poder dirigir el haz de luz directamente de una hacia la otra.

En el modo *casi-difuso*, el tipo de emisión es radial; esto es, la emisión se produce en todas direcciones. Para conseguir esto, lo que se hace es transmitir hacia distintas superficies reflectantes, las cuales redirigirán el haz de luz hacia las estaciones receptoras. En función de cómo sea esta superficie reflectante, se puede distinguir dos tipos de reflexión: pasiva y activa. En la reflexión pasiva, la superficie reflectante simplemente refleja la señal, debido a las cualidades reflexivas del material. En la reflexión activa, por el contrario, el medio reflectante no sólo refleja la señal, sino que además la amplifica.

El modo de emisión *difuso*, se diferencia del casi-difuso en que debe ser capaz de abarcar, mediante múltiples reflexiones, todo el recinto en el cual se encuentran las estaciones. Obviamente, esto requiere una potencia de emisión mayor que los dos modos anteriores, puesto que el número de rebotes incide directamente en el camino recorrido por la señal y las pérdidas aumentan.

2.3 EL CELULAR COMO MÓDEM EXTERNO

Para conocer el funcionamiento del teléfono celular como modem externo primero se debe conocer que es un módem y sus características.

2.3.1 MÓDEM

Un módem es un dispositivo que sirve para modular y demodular (en amplitud, frecuencia, fase u otro sistema) una señal llamada *portadora* mediante otra señal de entrada llamada *moduladora*. Se han usado módems desde los años 60 o antes del siglo XX, principalmente debido a que la transmisión directa de las señales electrónicas inteligibles, a largas distancias, no es eficiente, por ejemplo, para transmitir señales de audio por el aire, se requerirían antenas de gran tamaño (del orden de cientos de metros) para su correcta recepción.

2.3.1.1 Cómo funciona?

El modulador emite una señal denominada portadora. Generalmente, se trata de una simple señal eléctrica sinusoidal de mucha mayor frecuencia que la señal moduladora. La señal moduladora constituye la información que se prepara para una transmisión (un módem prepara la información para ser transmitida, pero no realiza la transmisión). La moduladora modifica alguna característica de la portadora (que es la acción de modular), de manera que se obtiene una señal, que incluye la información de la moduladora. Así el demodulador puede recuperar la señal moduladora original, quitando la portadora. Las características que se pueden modificar de la señal portadora son:

- Amplitud, dando lugar a una modulación de amplitud (AM/ASK).
- Frecuencia, dando lugar a una modulación de frecuencia (FM/FSK).
- Fase, dando lugar a una modulación de fase (PM/PSK)

También es posible una combinación de modulaciones o modulaciones más complejas como la modulación de amplitud en cuadratura.

2.3.1.2 Módems para PC

La distinción principal que se suele hacer es entre módems internos y módems externos, aunque recientemente han aparecido módems llamados "módems software", más conocidos como "winmódems" o "linuxmódems. También existen los módems para XDSL, RDSI, etc. y los que se usan para conectarse a través de cable coaxial de 75 ohms (cable módems).

- **Internos:** consisten en una tarjeta de expansión sobre la cual están dispuestos los diferentes componentes que forman el módem.
La principal ventaja de estos módems reside en su mayor integración con el ordenador, ya que no ocupan espacio sobre la mesa y reciben energía eléctrica directamente del propio ordenador. Además, suelen ser algo más baratos debido a que carecen de carcasa y transformador. Por el contrario, son algo más complejos de instalar y la información sobre su estado sólo puede obtenerse por software.
- **Externos:** similares a los anteriores, pero externos al ordenador o PDA. La ventaja de estos módems reside en su fácil transportabilidad entre ordenadores diferentes (algunos de ellos más fácilmente transportables y pequeños que otros), además de que es posible saber el estado del módem (marcando, con/sin línea, transmitiendo...) mediante los leds de estado que incorporan. Por el contrario, y obviamente, ocupan más espacio que los internos.
 - La conexión de los módems telefónicos con el ordenador se realiza generalmente mediante uno de los puertos serie tradicionales o COM.
 - Módems PC Card: son módems en forma de tarjeta, que se utilizaban en portátiles, antes de la llegada del USB.
 - Existen modelos para puerto USB, de conexión y configuración aún más sencillas, que no necesitan toma de corriente.
- **Módems software**, HSP (Host Signal Processor) o Winmódems: son módems generalmente internos, en los cuales se han eliminado varias piezas electrónicas (por ejemplo, chips especializados), de manera que el microprocesador del ordenador debe suplir su función mediante un programa.

Varios teléfonos celulares ofrecen la posibilidad de servir como un modem para telefonía móvil al ser conectados al computador principalmente vía USB. Pudiendo así acceder a internet sin la necesidad de una línea telefónica

convencional o brindando la alternativa de usar varias de las funciones que estos dispositivos poseen.

CAPÍTULO III

ESTUDIO DE HERRAMIENTAS, EQUIPOS Y DISPOSITIVOS EMPLEADOS

3.1 INTRODUCCIÓN

Para la creación de una aplicación que permita la integración de telefonía celular a través de mensajes de texto como medio de acceso a la información para la consulta de notas de los alumnos de la Escuela Politécnica del Ejército Sede Latacunga y el funcionamiento de la misma, se necesitarán de varios componentes, tanto de software como de hardware, los mismos que se describirán dentro de este capítulo

3.2 ESTUDIO DE HERRAMIENTAS DE DESARROLLO

La aplicación se desarrolló en su totalidad en el lenguaje de programación Java por ser un lenguaje de programación simple que elimina las características menos usadas de lenguajes como C++ por ser un lenguaje robusto, seguro y por tener una gran variedad de librerías de libre y gratuito uso, entre ellas algunas que permiten la comunicación del computador con diferentes dispositivos conectados a sus puertos, esta será una parte fundamental en el desarrollo y funcionamiento de la aplicación.

Se empleó el entorno de desarrollo integrado (IDE) Eclipse el cual es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido".

"Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent Azureus."¹⁴

¹⁴ [http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software))

A continuación se describen las principales características del lenguaje de programación Java y del Entorno de Desarrollo Integrado Eclipse.

Las características principales que ofrece Java respecto a cualquier otro lenguaje de programación, son:

Es Simple.- Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos. C++ es un lenguaje que adolece de falta de seguridad, pero C y C++ son lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje.

Java elimina muchas de las características de otros lenguajes como C++, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el garbage collector (reciclador de memoria dinámica). No es necesario preocuparse de liberar memoria, el reciclador se encarga de ello y como es un thread de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que reduce la fragmentación de la memoria. Java reduce en un 50% los errores más comunes de programación con lenguajes como C y C++ al eliminar muchas de las características de éstos, entre las que destacan:

- aritmética de punteros
- no existen referencias
- registros (struct)
- definición de tipos (typedef)
- macros (#define)
- necesidad de liberar memoria (free)

Aunque, en realidad, lo que hace es eliminar las palabras reservadas (struct, typedef), ya que las clases son algo parecido.

Además, el intérprete completo de Java que hay en este momento es muy pequeño, solamente ocupa 215 Kb de RAM.

Es Orientado A Objetos.- Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas, como en C++, clases y sus copias, instancias. Estas instancias, como en C++, necesitan ser construidas y destruidas en espacios de memoria.

Java incorpora funcionalidades inexistentes en C++ como por ejemplo, la resolución dinámica de métodos. En C++ se suele trabajar con librerías dinámicas (DLLs) que obligan a recompilar la aplicación cuando se retocan las funciones que se encuentran en su interior. Este inconveniente es resuelto por Java mediante una interfaz específica llamada RTTI (RunTime Type Identification) que define la interacción entre objetos excluyendo variables de instancias o implementación de métodos. Las clases en Java tienen una representación en el runtime que permite a los programadores interrogar por el tipo de clase y enlazar dinámicamente la clase con el resultado de la búsqueda.

Es Distribuido.- Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales.

La verdad es que Java en sí no es distribuido, sino que proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas, interactuando.

Es Robusto.- Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores lo antes posible en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria. También implementa los arrays

auténticos, en vez de listas enlazadas de punteros con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria resultado de punteros que señalan a zonas equivocadas. Estas características reducen drásticamente el tiempo de desarrollo de aplicaciones en Java.

Además, para asegurar el funcionamiento de la aplicación, realiza una verificación de los byte-codes, que son el resultado de la compilación de un programa Java. Es un código de máquina virtual que es interpretado por el intérprete Java. No es el código máquina directamente entendible por el hardware, pero ya ha pasado todas las fases del compilador: análisis de instrucciones, orden de operadores, etc., y ya tiene generada la pila de ejecución de órdenes.

Java proporciona, pues:

- Comprobación de punteros
- Comprobación de límites de arrays
- Excepciones
- Verificación de byte-codes

Es De Arquitectura Neutral.- Para establecer Java como parte integral de la red, el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. Cualquier máquina que tenga el sistema de ejecución (run-time) puede ejecutar ese código objeto, sin importar en modo alguno la máquina en que ha sido generado. Actualmente existen sistemas run-time para varios sistemas operativos.

El código fuente Java se "compila" a un código de bytes de alto nivel independiente de la máquina. Este código (byte-codes) está diseñado para ejecutarse en una máquina hipotética que es implementada por un sistema run-time, que sí es dependiente de la máquina.

Es Seguro.- La seguridad en Java tiene dos facetas. En el lenguaje, características como los punteros o el casting implícito que hacen los compiladores de C y C++ se eliminan para prevenir el acceso ilegal a la memoria. El lenguaje C, por ejemplo, tiene lagunas de seguridad importantes, como son los errores de alineación. Los programadores de C utilizan punteros en conjunción

con operaciones aritméticas. Esto le permite al programador que un puntero referencie a un lugar conocido de la memoria y pueda sumar (o restar) algún valor, para referirse a otro lugar de la memoria. Si otros programadores conocen nuestras estructuras de datos pueden extraer información confidencial de nuestro sistema. Otro tipo de ataque, es el Caballo de Troya. Se presenta un programa como una utilidad, resultando tener una funcionalidad destructiva.

El código Java pasa muchos tests antes de ejecutarse en una máquina. El código se pasa a través de un verificador de byte-codes que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal -código que falsea punteros, viola derechos de acceso sobre objetos o intenta cambiar el tipo o clase de un objeto.

Si los byte-codes pasan la verificación sin generar ningún mensaje de error, entonces sabemos que:

- El código no produce desbordamiento de operandos en la pila.
- El tipo de los parámetros de todos los códigos de operación son conocidos y correctos.
- No ha ocurrido ninguna conversión ilegal de datos, tal como convertir enteros en punteros.
- El acceso a los campos de un objeto se sabe que es legal: public, private, protected.
- No hay ningún intento de violar las reglas de acceso y seguridad establecidas.

El Cargador de Clases también ayuda a Java a mantener su seguridad, separando el espacio de nombres del sistema de ficheros local, del de los recursos procedentes de la red. Esto limita cualquier aplicación del tipo Caballo de Troya, ya que las clases se buscan primero entre las locales y luego entre las procedentes del exterior.

Las clases importadas de la red se almacenan en un espacio de nombres privado, asociado con el origen. Cuando una clase del espacio de nombres privado accede a otra clase, primero se busca en las clases predefinidas (del sistema local) y

luego en el espacio de nombres de la clase que hace la referencia. Esto imposibilita que una clase suplante a una predefinida.

“En resumen, las aplicaciones de Java resultan extremadamente seguras, ya que no acceden a zonas delicadas de memoria o de sistema, con lo cual evitan la interacción de ciertos virus.”¹⁵

Respecto a la seguridad del código fuente, no a la del lenguaje, JDK proporciona un desensamblador de byte-code que permite que cualquier programa pueda ser convertido a código fuente, lo que para el programador significa una vulnerabilidad total a su código. Utilizando java no se obtiene el código fuente original, pero sí desmonta el programa mostrando el algoritmo que se utiliza, que es lo realmente interesante.

Es Portable.- Más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo. Los enteros son siempre enteros y además, enteros de 32 bits en complemento a 2. Además, Java construye sus interfaces de usuario a través de un sistema abstracto de ventanas de forma que las ventanas puedan ser implantadas en entornos Unix, Pc o Mac.

Es Interpretado.- El intérprete Java (sistema run-time) puede ejecutar directamente el código objeto. Enlazar (linkar) un programa, normalmente, consume menos recursos que compilarlo por lo que los desarrolladores con Java pasarán más tiempo desarrollando y menos esperando por el ordenador. No obstante, el compilador actual del JDK es bastante lento. Por ahora, que todavía no hay compiladores específicos de Java para las diversas plataformas, Java es más lento que otros lenguajes de programación, como C++, ya que debe ser interpretado y no ejecutado como sucede en cualquier programa tradicional.

La verdad es que Java para conseguir ser un lenguaje independiente del sistema operativo y del procesador que incorpore la máquina utilizada, es tanto interpretado como compilado; y esto no es ningún contrasentido, el código fuente escrito con cualquier editor se compila generando el byte-code. Este código

¹⁵ http://www.wikilearning.com/tutorial/tutorial_de_java-caracteristicas_de_java/3938-5

intermedio es de muy bajo nivel, pero sin alcanzar las instrucciones máquina propia de cada plataforma.

El byte-code corresponde al 80% de las instrucciones de la aplicación. Ese mismo código es el que se puede ejecutar sobre cualquier plataforma; para ello hace falta el run-time, que sí es completamente dependiente de la máquina y del sistema operativo que interpreta dinámicamente el byte-code y añade el 20% de instrucciones que faltaban para su ejecución. Con este sistema es fácil crear aplicaciones multiplataforma, pero para ejecutarlas es necesario que exista el run-time correspondiente al sistema operativo utilizado.

Es Multithreaded.- Al ser multithreaded (multihilvanado, en mala traducción), Java permite muchas actividades simultáneas en un programa. Los threads (a veces llamados, procesos ligeros), son básicamente pequeños procesos o piezas independientes de un gran proceso. Al estar los threads construidos en el lenguaje, son más fáciles de usar y más robustos que sus homólogos en C o C++. El beneficio de ser multithreaded consiste en un mejor rendimiento interactivo y mejor comportamiento en tiempo real. Aunque el comportamiento en tiempo real está limitado a las capacidades del sistema operativo subyacente (Unix, Windows, etc.), aún supera a los entornos de flujo único de programa (single-threaded) tanto en facilidad de desarrollo como en rendimiento.

Es Dinámico.- Java se beneficia todo lo posible de la tecnología orientada a objetos. Java no intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución. Las librerías nuevas o actualizadas no paralizarán las aplicaciones actuales (siempre que mantengan el API anterior). Java también simplifica el uso de protocolos nuevos o actualizados. Si su sistema ejecuta una aplicación Java sobre la red y encuentra una pieza de la aplicación que no sabe manejar, tal como se ha explicado en párrafos anteriores, Java es capaz de traer automáticamente cualquiera de esas piezas que el sistema necesita para funcionar.

Java, para evitar que los módulos de byte-codes o los objetos o nuevas clases, sean llamados de la red cada vez que se necesiten, implementa las opciones de persistencia para que no se eliminen cuando se limpie la caché de la máquina.

3.3 ECLIPSE

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación.

3.3.1 Arquitectura

Eclipse, es una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

La base para Eclipse es la Plataforma de cliente enriquecido (del Inglés Rich Client Platform RCP). Los siguientes componentes constituyen la plataforma de cliente enriquecido:

- Plataforma principal - inicio de Eclipse, ejecución de plugins.
- OSGi - una plataforma para bundling estándar.
- El Standard Widget Toolkit (SWT) - Un widget toolkit portable.
- JFace - manejo de archivos, manejo de texto, editores de texto.
- El Workbench de Eclipse - vistas, editores, perspectivas, asistentes.

Los widgets de Eclipse están implementados por una herramienta de widget para Java llamada SWT, a diferencia de la mayoría de las aplicaciones Java, que usan las opciones estándar Abstract Window Toolkit (AWT) o Swing. La interfaz de usuario de Eclipse también tiene una capa GUI intermedia llamada JFace, la cual simplifica la construcción de aplicaciones basada en SWT.

El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (en inglés *plug-in*) para proporcionar toda su funcionalidad al frente de la plataforma de cliente rico, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas las necesite el usuario o no. Este

mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente permitirle a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python, permite a Eclipse trabajar con lenguajes para procesado de texto como LaTeX, aplicaciones en red como Telnet y Sistema de gestión de base de datos. La arquitectura plugin permite escribir cualquier extensión deseada en el ambiente, como sería Gestión de la configuración.

Se provee soporte para Java y CVS en el SDK de Eclipse y no tiene por qué ser usado únicamente para soportar otros lenguajes de programación.

La definición que da el proyecto Eclipse acerca de su software es: "*una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular*".

En cuanto a las aplicaciones clientes, eclipse provee al programador con frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones web, etc.

El SDK de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador de Java interno y un modelo completo de los archivos fuente de Java. Esto permite técnicas avanzadas de refactorización y análisis de código. El IDE también hace uso de un espacio de trabajo, en este caso un grupo de metadata en un espacio para archivos plano, permitiendo modificaciones externas a los archivos en tanto se refresque el espacio de trabajo correspondiente.

3.4 ESTUDIO DE EQUIPOS Y DISPOSITIVOS A UTILIZARSE

Los equipos y dispositivos que se utilizaron para el desarrollo y funcionamiento de la aplicación son un computador, un teléfono celular y un cable para conectar el teléfono celular al computador; los mismos que son detallados a continuación:

- Computador:

Para el desarrollo de la aplicación se utilizó un computador con las siguientes características:

Característica	Descripción
Computador	Hp Pavilion Dv 6235us
Procesador	Core Duo 1.73 Ghz
Memoria Ram	1 Gb
Disco Duro	120 Gb
Puertos USB	3 Usb 2.0
Tarjeta de Red (LAN)	Intel(R) Pro/100 Ve Network
Tarjeta de Red (Wireless)	Intel(R) PRO/Wireless 3945ABG Network
Monitor	15.4 pulgadas
Sistema Operativo	Windows Vista Home Premium

Tabla 3.1: Características del computador utilizado

- Teléfono Celular

Se empleó un teléfono Motorola L6 por poseer conexión USB con el computador, se carga mediante la conexión con cable USB al computador y la capacidad de funcionar como modem. A continuación se detallan las principales características del mismo:

Característica	Descripción
Tecnología	GSM
Redes	850 / 900 / 1800 / 1900
Tecnología EDGE	GSM
Tiempo de Espera	310 horas
Tiempo de Conversación	4 horas
Conectividad	Bluethooth, GPRS, Modem, USB, Wap 2.0
Características de Mensajes	EMS, SMS, MMS, Email
Batería	Li-Ion 820 mAh
Capacidad Almacenamiento	5 Mb

Tabla 3.2: Características del teléfono celular utilizado

- Cable USB:

Se utilizó un cable que en uno de sus extremos posee un conector mini USB y en el otro extremo un conector USB estándar, éste se conecta en el teléfono celular y en el computador respectivamente.

Este cable permite la transmisión de datos entre el teléfono celular y el computador a la vez que proporcionara energía para que el teléfono celular se cargue mientras esté conectado al computador.

3.5 ESTUDIO DE SISTEMAS OPERATIVOS Y DRIVERS

En este apartado se describen las características principales del sistema operativo en el cual se desarrolló la aplicación y los drivers necesarios para el funcionamiento de la aplicación.

3.5.1 SISTEMA OPERATIVO

Como se mencionó anteriormente se utilizó un sistema operativo Windows Vista, el mismo que tiene las siguientes características principales:

Windows Vista es una línea de sistemas operativos desarrollada por Microsoft para ser usada en ordenadores de sobremesa, ordenadores portátiles, Tablet PC y Centros multimedia.

Novedades y Características:

- Internet Explorer 7: Viene incorporado con Windows Vista el cual incorpora varias mejoras como la navegación con pestañas y la vista Quick Tabs que muestra vistas en miniatura de las páginas abiertas.
- Windows Vista es el primer sistema operativo de Microsoft concebido para garantizar una compatibilidad total con EFI (Extensible Firmware Interface), la tecnología llamada a reemplazar a las arcaicas BIOS que desde hace más de dos décadas han formado parte indisoluble de los ordenadores personales, por lo tanto no empleará MBR (Master Boot Record) sino GPT (GUID Partition Table)
- Ventanas dibujadas con gráficos vectoriales usando XAML y DirectX. Para ello se utilizaría una nueva API, llamada *Windows Presentation Foundation*,

cuyo nombre en código es Avalon, que requiere una tarjeta gráfica con aceleración 3D compatible con DirectX.

- *WinFX*, una API orientada a reemplazar la API actual llamada Win32 está, junto con Avalon e Indigo, son los pilares de Windows Vista.
- Capacidad nativa para grabar DVD.
- Una interfaz de línea de comando denominada *Windows PowerShell*, que finalmente se ofreció como una descarga independiente para Windows Vista y Windows XP SP2.
- La utilidad de restauración del sistema ha sido actualizada e implementada como herramienta de inicio de sesión, facilitando así el "rescate" del sistema.
- Un sistema unificado de comunicaciones llamado Windows Communication Foundation, cuyo nombre en código es Indigo.
- Un sistema antispyware denominado Windows Defender.
- Añade al firewall de sistema la capacidad de bloquear conexiones que salen del sistema sin previa autorización.
- Se incluye Windows ReadyBoost que es una tecnología de cache de disco incluida por primera vez en el sistema operativo Windows Vista. Su objetivo es hacer más veloces a aquellos computadores que se ejecutan con el mencionado sistema operativo mediante pendrives, tarjetas SD, CompactFlash o similares.
- Windows Vista Starter: esta edición está diseñada para mercados en desarrollo, especialmente para computadoras de bajo costo.

Los requerimientos mínimos para instalar Windows Vista son:

Requerimientos Mínimos	
Procesadores	800 MHz 32-bit (x86) o 64-bit (x64)
Memoria	512 MB
GPU	SVGA (800x600)
Memoria Gráfica	32 MB
Disco Duro	20 GB
Espacio libre en disco	15 GB
Disco óptico	CD-ROM drive

Tabla 3.3: Requerimientos mínimos Windows Vista

3.5.2 DRIVERS

Se empleó el driver para el teléfono celular Motorola L6 el mismo que permite conectar el teléfono al computador a través de un cable USB y así poder ocupar la función de modem que es necesaria para el funcionamiento de la aplicación.

El driver tiene un tamaño de 3.41 MB y posee una licencia de tipo freeware (Gratis).

A continuación se describen algunos aspectos importantes acerca de este driver:

Requerimientos de “Drivers para Motorola L6”

- Windows 2000/XP/Vista.
- Teléfono celular Motorola L6.
- Procesador Pentium 466 MHz.
- 120 MB de espacio libre en el disco duro.
- 64 MB de memoria RAM.

CAPÍTULO IV

DESARROLLO DE LA APLICACIÓN

4.1 DESCRIPCIÓN DE LA APLICACIÓN

La aplicación permite que los alumnos de la Escuela Politécnica del Ejército sede Latacunga puedan consultar las notas de una materia específica a través del envío de un mensaje de texto desde cualquier celular movistar, para esto deberán enviar en el mensaje de texto su número de cédula y el nombre de la materia que deseen consultar las notas, cada parámetro separado por una coma y en el orden mencionado.

El mensaje será enviado a un teléfono celular, el mismo que estará conectado a través de un cable USB a un computador, que deberá poseer acceso a la base de datos de la Escuela Politécnica del Ejército sede Latacunga con los siguientes campos:

- Número de cédula de los alumnos.
- Nombres y apellidos de los alumnos.
- Materias en las que se encuentran matriculados en el período en vigencia.
- Códigos de las materias en las que se encuentran matriculados en el período en vigencia.
- Notas de las materias en las que se encuentran matriculados en el período vigente.

La aplicación abrirá el puerto de comunicaciones en el cual se encuentra conectado el teléfono celular que recibe los mensajes que envían los alumnos, buscará cada cierto tiempo los mensajes nuevos contenidos en la bandeja de entrada, para luego analizar las cadenas de texto y de acuerdo a estas cadenas de texto realizar la consulta a la base de datos con los parámetros requeridos; una vez realizada la consulta a la base de datos se creará un mensaje con los resultado de dicha consulta y se enviará al remitente del mensaje.

4.2 ESPECIFICACIÓN DE REQUISITOS

4.2.1 INTRODUCCIÓN

Este apartado del documento es una especificación de requisitos de software (ERS) para la aplicación que permitirá la consulta de notas a través de mensajes de texto de los alumnos de la Escuela Politécnica del Ejército sede Latacunga. Esta especificación de requisitos se ha estructurado tomando como base las directrices dadas por el standard “IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830 1998”

4.2.1.1 Propósito

El objeto del presente apartado del documento es definir de manera clara y precisa todas las funcionalidades y restricciones de la aplicación a ser desarrollada, la cual deberá: permitir la consulta de notas de una materia de los alumnos de la Escuela Politécnica del Ejército sede Latacunga a través de mensajes de texto, lo que implica que deberá recibir los mensajes, analizarlos, realizar la consulta en la base de datos y enviar el resultado de la consulta. Todo ello se realizará en forma automática sin necesidad de que una persona opere la aplicación.

4.2.1.2 Ámbito de la aplicación

Para el presente documento la aplicación tendrá el nombre de “TesisSMS”.

El motor que impulsa la realización de la presente aplicación es la decisión de realizar un proyecto que beneficie a todos los integrantes de Escuela Politécnica del Ejército sede Latacunga.

La situación de partida es una en la que existe una base de datos con todos los parámetros requeridos para el funcionamiento de la aplicación y una aplicación web que permite realizar la consulta de notas de los alumnos de la Escuela Politécnica del Ejército sede Latacunga.

Esta aplicación podría convertirse en una alternativa que contribuya a que no se produzca un saturamiento del sistema existente.

4.2.1.3 Definición de acrónimos y abreviaturas

4.2.1.3.1 Definiciones

Alumno	Persona que realiza una consulta a través de la aplicación
Materia	Asignatura de la cual se desea consultar las notas
Notas	Calificaciones obtenidas por los alumnos en las distintas materia

Tabla 4.1: Definiciones

4.2.1.3.2 Acrónimos

ERS	Especificación de requisitos de sistema
ARS	Análisis de requisitos del sistema

Tabla 4.2: Acrónimos

4.2.1.3.3 Abreviaturas

TesisSms	Aplicación para la consulta de notas de los alumnos de la ESPEL
ESPEL	Escuela Politécnica del Ejercito Sede Latacunga

Tabla 4.3: Abreviaturas

4.2.1.4 Referencias

IEEE Recommended Practice for Software Requeriments Specification. ANSI/IEEE std 830, 1998.

4.2.1.5 Visión general del documento

Este documento consta de tres secciones, esta sección es la introducción y proporciona una visión general de la ERS. En la sección dos se da una descripción general de la aplicación, con el fin de conocer las principales funciones que debe realizar, los datos asociados y los factores, restricciones y dependencias que afectan al desarrollo, en la sección tres se detallan los requisitos que debe satisfacer el sistema.

4.2.2 DESCRIPCIÓN GENERAL

En esta sección se presenta una descripción general de la aplicación, con el fin de conocer las principales funciones que debe realizar, los datos asociados, las restricciones impuestas, y cualquier factor que pueda afectar al desarrollo del mismo.

4.2.2.1 Perspectiva del producto

El sistema en esta versión, no interactuará con ningún otro sistema informático.

4.2.2.1.1 Funciones de la aplicación

En términos generales la aplicación deberá permitir la consulta de notas de los alumnos de la ESPEL a través de mensajes de texto por lo que deberá cumplir con las siguientes tareas:

- Conexión con un teléfono celular
- Recibir mensajes del usuario
- Descargar mensajes de la bandeja de entrada
- Analizar el mensaje de texto
- Realizar consulta en la base de datos
- Estructurar un mensaje de texto
- Registrar los eventos
- Enviar el mensaje al usuario

Conexión con un teléfono celular

La aplicación deberá permitir la conexión con un teléfono celular que reciba y envíe los mensajes de texto para la consulta de notas, tendrá que establecer todas las conexiones con el puerto de comunicaciones al cual el teléfono celular se encuentre conectado.

Descargar mensajes de la bandeja de entrada

Una vez establecida la conexión con el teléfono celular la aplicación deberá verificar constantemente la existencia de mensajes de texto nuevos en la bandeja de entrada para descargarlos.

Analizar el mensaje de texto

Luego de descargados los mensajes de texto desde el teléfono celular se procederá a analizar el contenido del mismo, dicho análisis obtendrá como resultado los parámetros para la consulta en la base de datos.

Realizar consulta en la base de datos

Con los parámetros obtenidos resultantes del análisis del mensaje se realizara una consulta en la base de datos.

Estructurar un mensaje de texto

Realizada la consulta en la base de datos se procederá a estructurar un mensaje de texto con los resultados que arroje dicha consulta.

Colocar el mensaje de texto en la bandeja de salida

El mensaje de texto se lo colocara en la bandeja de salida del teléfono celular, esto permitirá que el mensaje sea enviado al destinatario.

4.2.2.2 Características de los usuarios

La aplicación no poseerá una interfaz de usuario, siendo la única forma de interactuar con el sistema el envío y recepción de mensajes de texto desde y hacia un teléfono celular.

4.2.2.3 Restricciones

- La aplicación deberá permitir la consulta de notas de una sola materia por mensaje enviado.
- Para la consulta de notas a través de la aplicación se fijara una operadora, lo que conlleva a que los usuarios deben enviar mensajes solamente de la operadora escogida.

4.2.2.3.1 Suposiciones

Los usuarios deberán conocer su número de cédula y el nombre completo de la materia que desean consultar, para poder realizar la consulta de notas a través de mensajes de texto.

4.2.2.3.2 Dependencias

La aplicación a desarrollar no tiene dependencia respecto a otros sistemas, los únicos factores de los que depende son un teléfono celular y la base de datos de la Escuela Politécnica del Ejército sede Latacunga.

4.2.3 REQUISITOS ESPECÍFICOS

En este apartado se describirá los requisitos funcionales que deberán ser satisfechos por la aplicación.

4.2.3.1 Requisitos Funcionales

4.2.3.1.1 Conexión con un teléfono celular

La aplicación deberá permitir:

- Req(01) Acceder al puerto de comunicaciones al que se encuentre conectado el teléfono celular.

4.2.3.1.2 Recibir mensajes del usuario

- Req(02) Recibir mensajes con la cédula y la materia que desean consultar los usuarios.

4.2.3.1.3 Descargar mensajes de la bandeja de entrada

La aplicación deberá permitir:

- Req(03) Tener acceso a la bandeja de entrada del teléfono celular.
- Req(04) Descargar los mensajes nuevos de la bandeja de entrada del teléfono celular.

4.2.3.1.4 Analizar el mensaje de texto

La aplicación deberá permitir:

- Req(05) Identificar el número del teléfono celular que envía el mensaje de texto.
- Req(06) Identificar el número de cédula contenido en el mensaje de texto.
- Req(07) Identificar el dígito separador dentro del mensaje de texto.
- Req(08) Identificar el nombre de la materia a consultar enviado en el mensaje de texto.

4.2.3.1.5 Realizar consulta en la base de datos

La aplicación deberá permitir:

- Req(09) Validar que el número de cédula exista en la base de datos.
- Req(10) Verificar que la materia sea correcta y pertenezca al número de cédula enviado.
- Req(11) Consultar las notas de la materia enviada.

4.2.3.1.6 Estructurar un mensaje de texto

La aplicación deberá permitir:

- Req(12) Estructurar un mensaje de texto con las notas resultantes de la consulta a la base de datos.

4.2.3.1.7 Registrar eventos

La aplicación deberá permitir:

- Req(13) Crear un archivo de texto.
- Req(14) Registrar los eventos en el archivo de texto.

4.2.3.1.8 Enviar el mensaje al usuario

La aplicación deberá permitir:

- Req(15) Colocar el mensaje en la bandeja de salida del teléfono celular.
- Req(16) Enviar mensajes a través del teléfono celular.

4.2.3.2 Requisitos de interfaces externas

4.2.3.2.1 Interfaces de usuario

La aplicación no poseerá una interfaz de usuario.

4.2.3.2.2 Interfaces de software

No habrá ninguna interfaz software con sistemas externos.

4.2.3.2.3 Interfaces de comunicación

La aplicación estará instalada directamente en donde se encuentre la base de datos o donde se pueda tener acceso a ella, a través de red o localmente.

4.2.3.3 Requisitos de rendimiento

No se han definido.

4.2.3.4 Requisitos de desarrollo

El ciclo de vida para desarrollar el producto es el secuencial básico, o en cascada.

4.2.3.5 Requisitos tecnológicos

La aplicación podrá ejecutarse sobre un PC con la siguiente configuración o superior:

- Procesador Pentium IV
- Memoria Ram 256 Mb
- Disco duro 5 Gb
- 1 Puerto USB 2.0

El sistema operativo en la que se debe ejecutar la aplicación será Windows 98 o superior.

Para el acceso a la base de datos se utilizará ODBC.

4.2.3.6 Atributos

4.2.3.6.1 Seguridad

Cuando un usuario de la aplicación intente acceder a ella a través de mensajes de texto, deberá ingresar su identificación y la materia, las cuales están registradas

previamente en la base de datos. Si el identificador introducido junto con la materia no corresponde, se le indicará con un mensaje de error.

4.3 MODELAMIENTO

Para el moldeamiento de la aplicación se empleó UML.

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

Los principales diagramas que usa UML son:

- Diagramas de Casos de Uso
- Diagramas de Secuencia
- Diagramas de Estado
- Diagramas de Clases

Los mismos que se emplearan a continuación.

4.3.1 DIAGRAMA DE CASOS DE USO

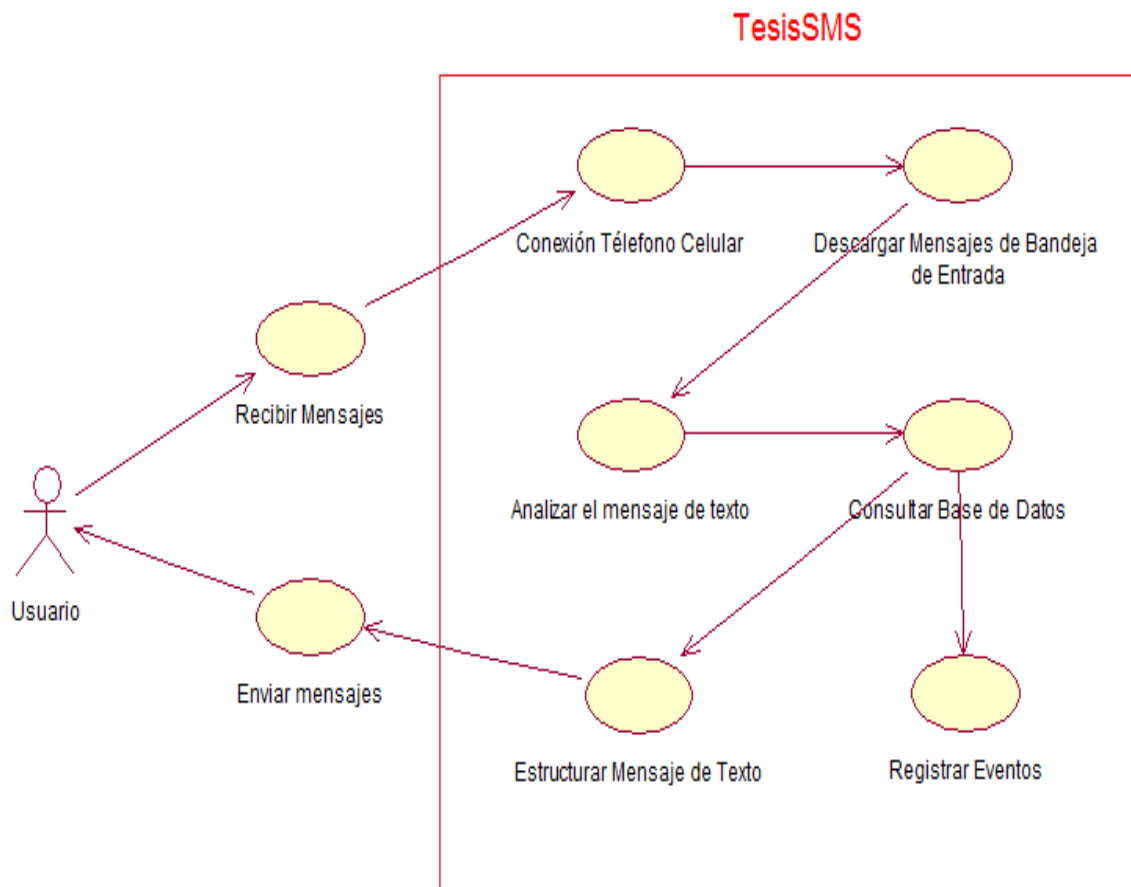


Diagrama 4.1: Diagrama de casos de uso

4.3.1.1 Definición de casos de uso de alto nivel

Caso de Uso: Recibir mensajes	
Actores: Teléfono celular	
Propósito: Recibir mensajes con los datos del usuario	
Resumen: El usuario envía en un mensaje de texto su cedula y la materia a consultar y lo envía al teléfono celular que funciona como módem de la aplicación.	
Tipo: Primario y Esencial	
Curso Normal de los Eventos	
Acción del Actor	Respuesta de la Aplicación
1. Este caso de uso comienza cuando el usuario manda un mensaje de texto con sus datos y el nombre de la materia a consultar.	
	2. Recibe el mensaje de texto.
Cursos Alternos	
<ul style="list-style-type: none"> • Línea 1: El usuario envía el mensaje a un número erróneo 	

Tabla 4.4: Caso de uso recibir mensajes

Diagrama de secuencia

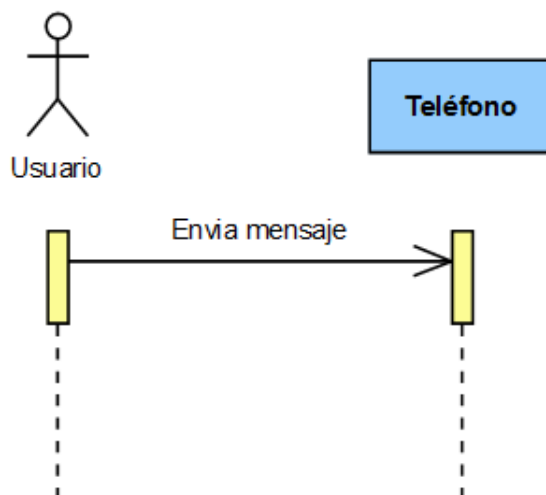


Diagrama 4.2: Diagrama de secuencia recibir mensajes

Caso de Uso: Conexión con el teléfono celular	
Actores: Desarrollador, teléfono, PC	
Propósito: Detectar y utilizar un teléfono celular como módem	
Resumen: Se conecta el teléfono celular al PC y a través de una librería de comunicaciones la aplicación accede a él para hacer uso del mismo.	
Tipo: Primario y Esencial	
Curso Normal de los Eventos	
Acción del Actor	Respuesta de la Aplicación
1. Este caso de uso comienza al conectar el teléfono celular al PC	
	2. Detecta el teléfono celular
	3. Solicita la instalación de drivers del teléfono
4. El usuario instala el driver	
	5. Reconoce el teléfono celular como módem
Cursos Alternos	
<ul style="list-style-type: none"> • Línea 2: No detecta el teléfono celular • Línea 4: No se instala el driver o es incorrecto. 	

Tabla 4.5: Caso de uso conexión con el teléfono celular

Diagrama de secuencia

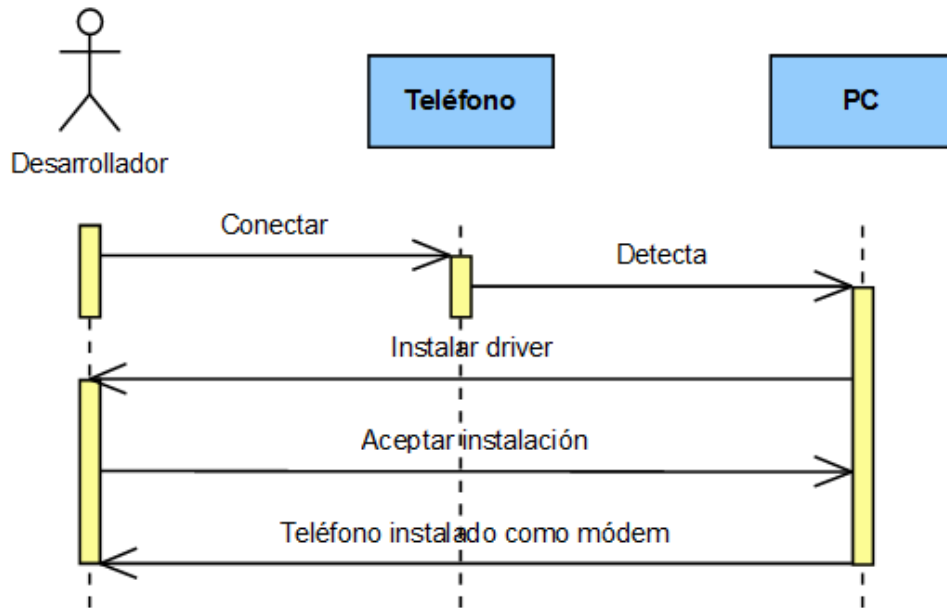


Diagrama 4.3: Diagrama de secuencia conexión con el teléfono celular

Caso de Uso: Descargar mensajes de la bandeja de entrada	
Actores: Desarrollador, Aplicación, teléfono	
Propósito: Descargar los mensajes nuevos que se encuentren en la bandeja de entrada del teléfono celular	
Resumen: Se accede a la bandeja de entrada del teléfono celular y se verifica si existen nuevos mensajes para descargarlos y posteriormente analizarlos.	
Tipo: Primario y Esencial	
Curso Normal de los Eventos	
Acción del Actor	Respuesta de la Aplicación
1. Este caso de uso comienza al ejecutar la aplicación	
	2. Accede a la bandeja de entrada del teléfono celular
	3. Busca si existen mensajes nuevos
	4. Descarga los mensajes nuevos

Cursos Alternos
<ul style="list-style-type: none"> • Línea 2: No existen mensajes nuevos

Tabla 4.6: Caso de uso descargar mensajes de la bandeja de entrada

Diagrama de secuencia

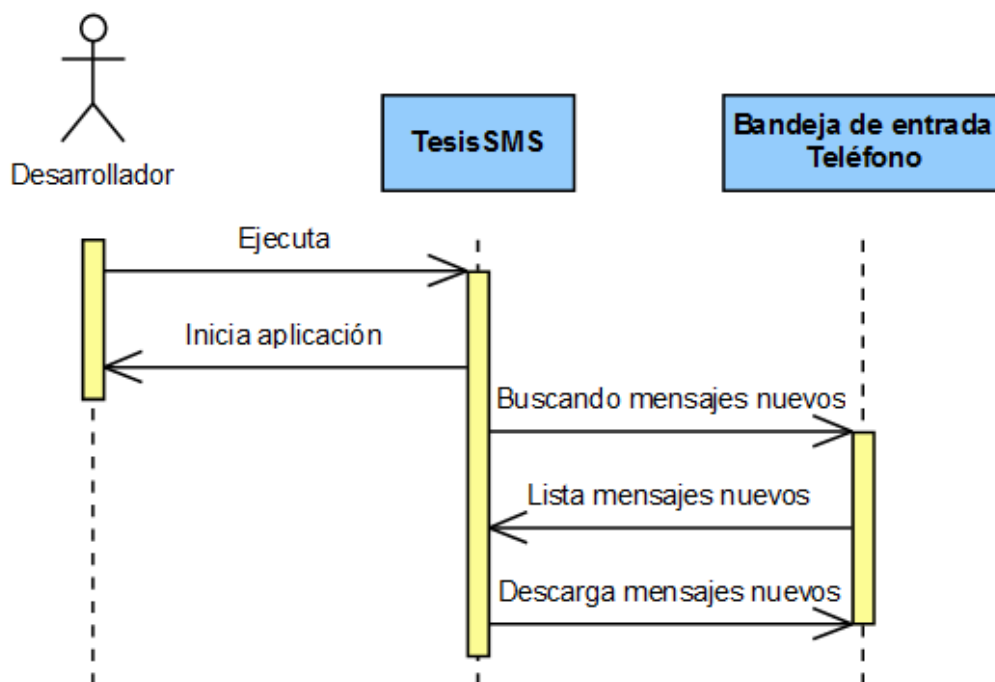


Diagrama 4.4: Diagrama de secuencia descargar mensajes de la bandeja de entrada

Caso de Uso: Analizar el mensaje de texto	
Actores: Aplicación	
Propósito: Descomponer y analizar el mensaje de texto	
Resumen: Se descompone el mensaje y se obtiene el texto del mismo, luego se separa la cedula y el nombre de la materia.	
Tipo: Primario y Esencial	
Curso Normal de los Eventos	
Acción del Actor	Respuesta de la Aplicación
	<ol style="list-style-type: none"> 1. Este caso de uso empieza cuando se descarga un mensaje

	nuevo de la bandeja de entrada
	2. Obtiene la cabecera del mensaje
	3. Obtiene el texto del mensaje
	4. Del texto del mensaje separa la cédula y el nombre de la materia.
Cursos Alternos	
<ul style="list-style-type: none"> • Línea 3: El mensaje esta vacio 	
<ul style="list-style-type: none"> • Línea 4: No existen cédula o materia 	

Tabla 4.7: Caso de uso analizar el mensaje de texto

Diagrama de secuencia

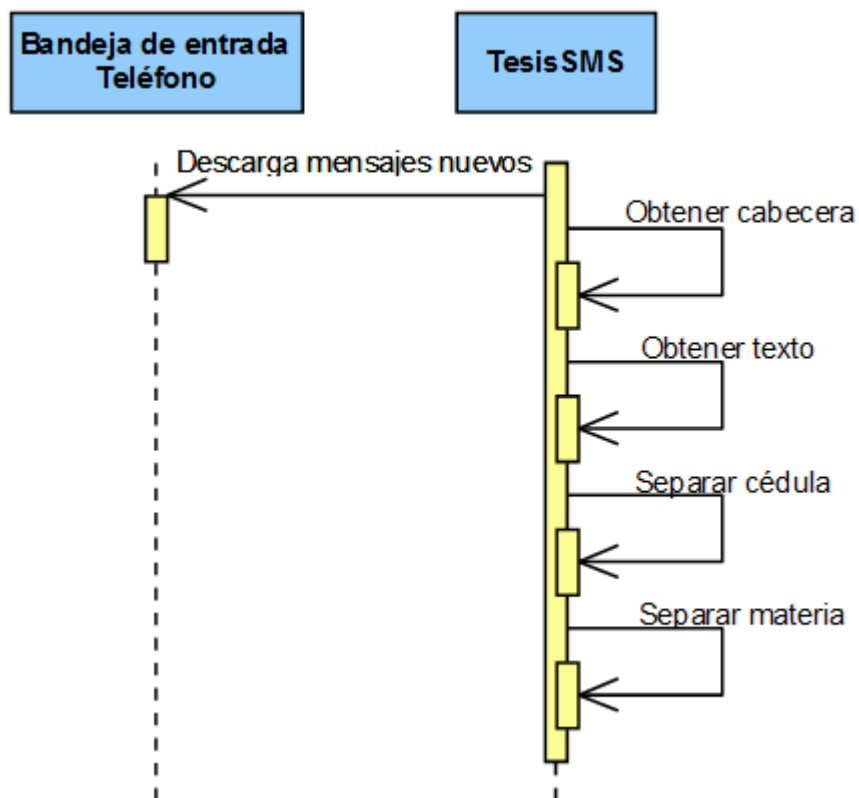


Diagrama 4.5: Diagrama de secuencia analizar el mensaje de texto

Caso de Uso: Consultar base de datos	
Actores: Aplicación, base de datos	
Propósito: Realizar una consulta a la base de datos	
Resumen: Se realiza una consulta a la base de datos con los parámetros enviados en el mensaje de texto	
Tipo: Primario y Esencial	
Curso Normal de los Eventos	
Acción del Actor	Respuesta de la Aplicación
	1. Este caso de uso comienza cuando se ha terminado el proceso de análisis del mensaje
	2. Se conecta a la base de datos
	3. Crea una sentencia SQL
	4. Ejecuta la sentencia.
	5. Obtiene el resultado de la consulta.
Cursos Alternos	
<ul style="list-style-type: none"> • Línea 2: Error de conexión con la base de datos 	
<ul style="list-style-type: none"> • Línea 3: Error en la sentencia SQL 	
<ul style="list-style-type: none"> • Línea 5: No existen registros 	

Tabla 4.8: Caso de uso consultar base de datos

Diagrama de secuencia

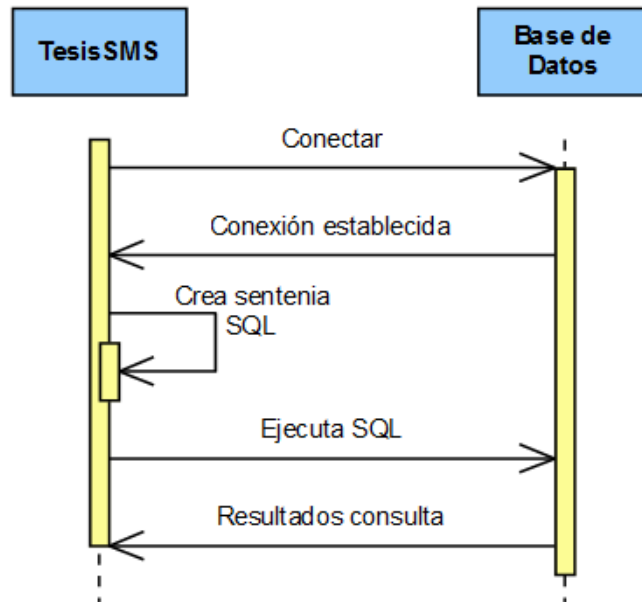


Diagrama 4.6: Diagrama de secuencia consultar base de datos

Caso de Uso: Registrar eventos	
Actores: Aplicación	
Propósito: Registrar en un archivo de texto los eventos que ocurran en la aplicación	
Resumen: Se crea un archivo de texto y se registran en él algunos eventos que ocurren en el funcionamiento de la aplicación	
Tipo: Secundario	
Curso Normal de los Eventos	
Acción del Actor	Respuesta de la Aplicación
	1. Este caso de uso empieza cuando se ejecuta la aplicación
	2. Crea un archivo de texto en una ubicación específica
	3. Registra los eventos
Cursos Alternos	
<ul style="list-style-type: none"> • Línea 2: El archivo de texto ya existe 	

Tabla 4.9: Caso de uso registrar eventos

Diagrama de secuencia

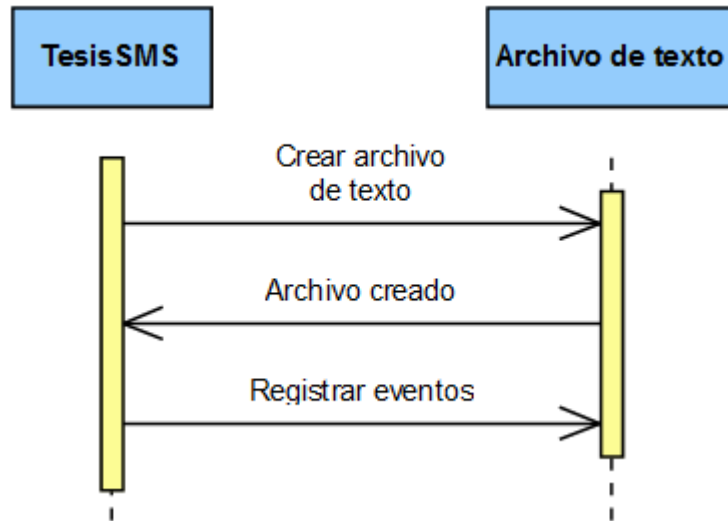


Diagrama 4.7: Diagrama de secuencia registrar eventos

Caso de Uso: Estructurar un mensaje de texto	
Actores: Aplicación, base de datos	
Propósito: Estructurar un mensaje de texto con el resultado de la consulta	
Resumen: Se estructura un mensaje de texto con el resultado de la consulta y con el número del celular que envió la petición de consulta	
Tipo: Primario y Esencial	
Curso Normal de los Eventos	
Acción del Actor	Respuesta de la Aplicación
	1. Este caso de uso inicia cuando finaliza la consulta a la base de datos
	2. Llama a los resultados de la consulta
	3. Concatena los resultados
	4. Declara el número para el envío.
Cursos Alternos	
<ul style="list-style-type: none"> • Línea 2: No existen resultados 	

Tabla 4.10: Caso de uso estructurar un mensaje de texto

Diagrama de secuencia

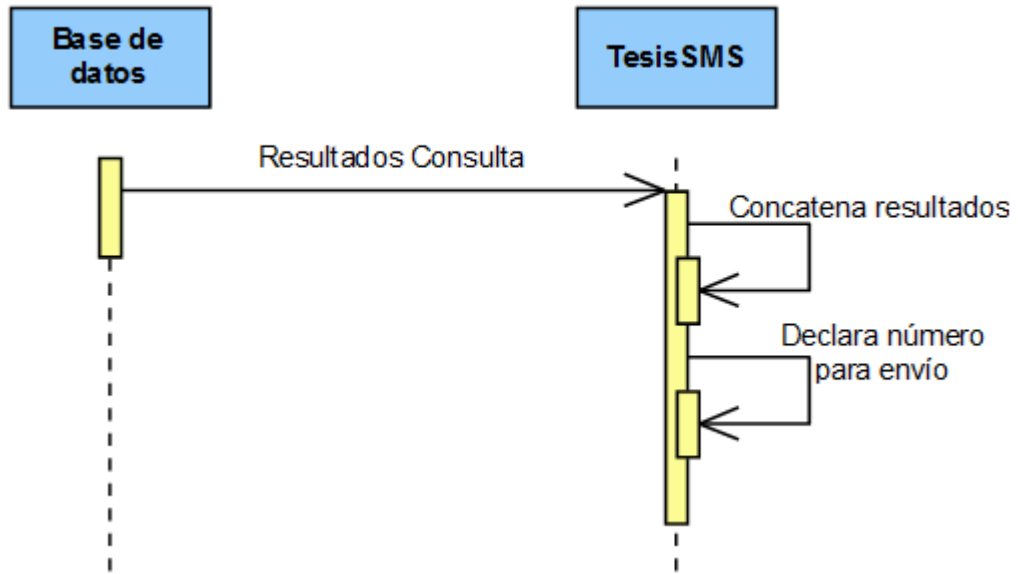


Diagrama 4.8: Diagrama de secuencia estructurar un mensaje de texto

Caso de Uso: Enviar mensajes	
Actores: Aplicación, teléfono	
Propósito: Enviar mensajes con los resultados de las consultas	
Resumen: Se coloca el mensaje ya estructurado en la bandeja de salida para luego enviarlo.	
Tipo: Primario y Esencial	
Curso Normal de los Eventos	
Acción del Actor	Respuesta de la Aplicación
	1. Este caso de uso comienza cuando se ha estructurado el mensaje de texto con el resultado de la consulta.
	2. Coloca el mensaje en la bandeja de salida
3. Envía el mensaje	

Tabla 4.11: Caso de uso enviar mensajes

Diagrama de secuencia

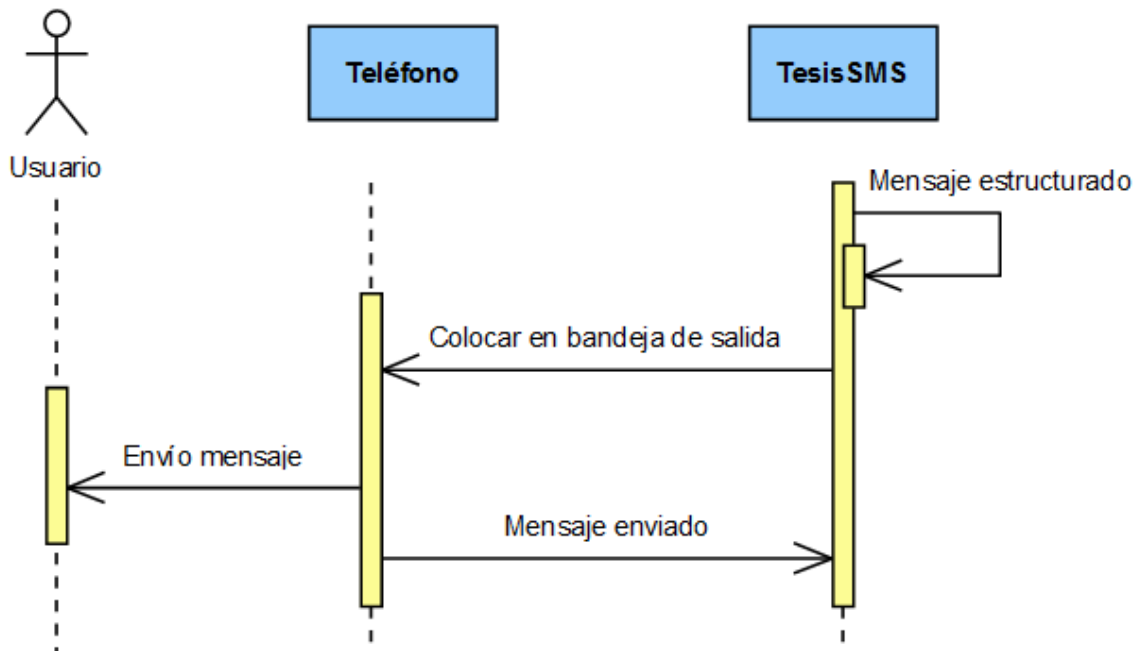


Diagrama 4.9: Diagrama de secuencia enviar mensajes

4.3.2 DIAGRAMAS DE ESTADO

Recibir Mensajes

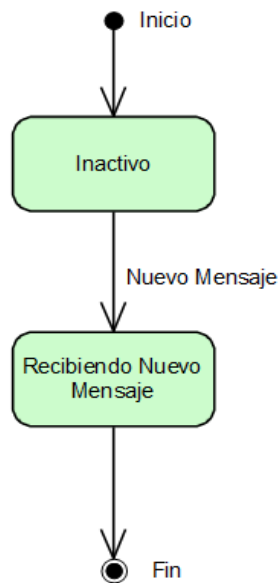


Diagrama 4.10: Diagrama de estado recibir mensajes

Conexión teléfono celular

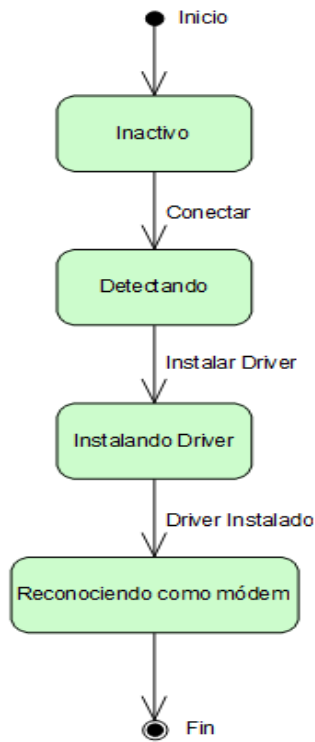


Diagrama 4.11: Diagrama de estado conexión con el teléfono celular

Descargar mensajes de bandeja de entrada

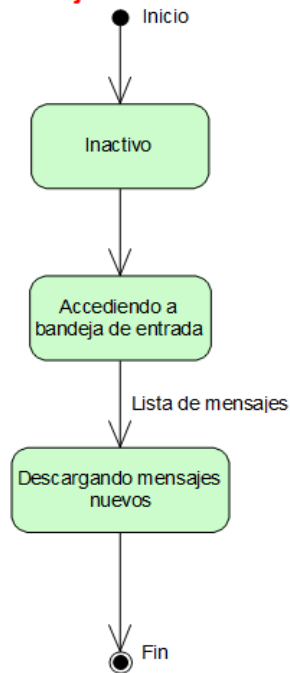


Diagrama 4.12: Diagrama de estado descargar mensajes de bandeja de entrada

Analizar el mensaje de texto

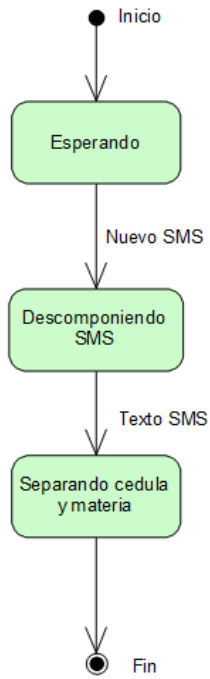


Diagrama 4.13: Diagrama de estado analizar el mensaje de texto

Consultar Base de Datos

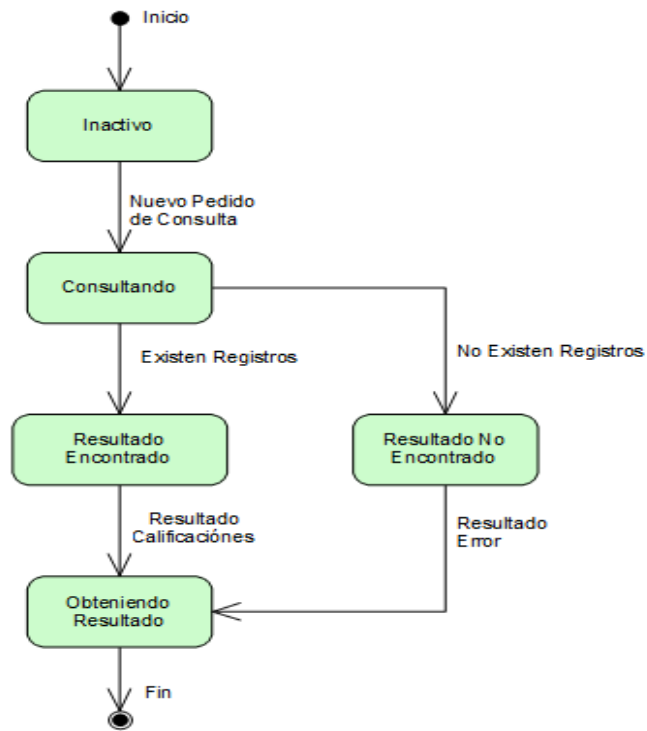


Diagrama 4.14: Diagrama de estado consultar base de datos

Registrar Eventos



Diagrama 4.15: Diagrama de estado registrar eventos

Estructurar mensaje de texto

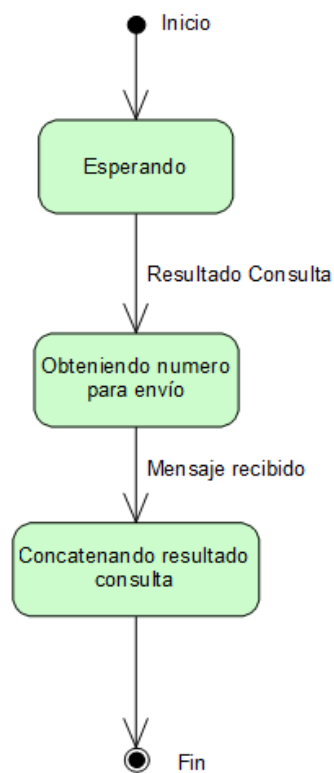


Diagrama 4.16: Diagrama de estado estructurar mensaje de texto

Enviar mensaje

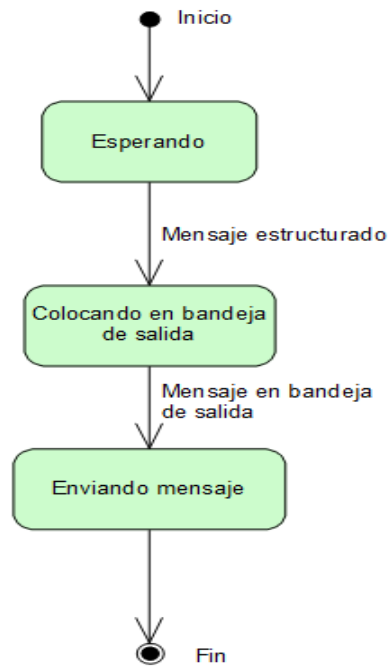


Diagrama 4.17: Diagrama de estado enviar mensaje

4.3.3 DIAGRAMA DE CLASES

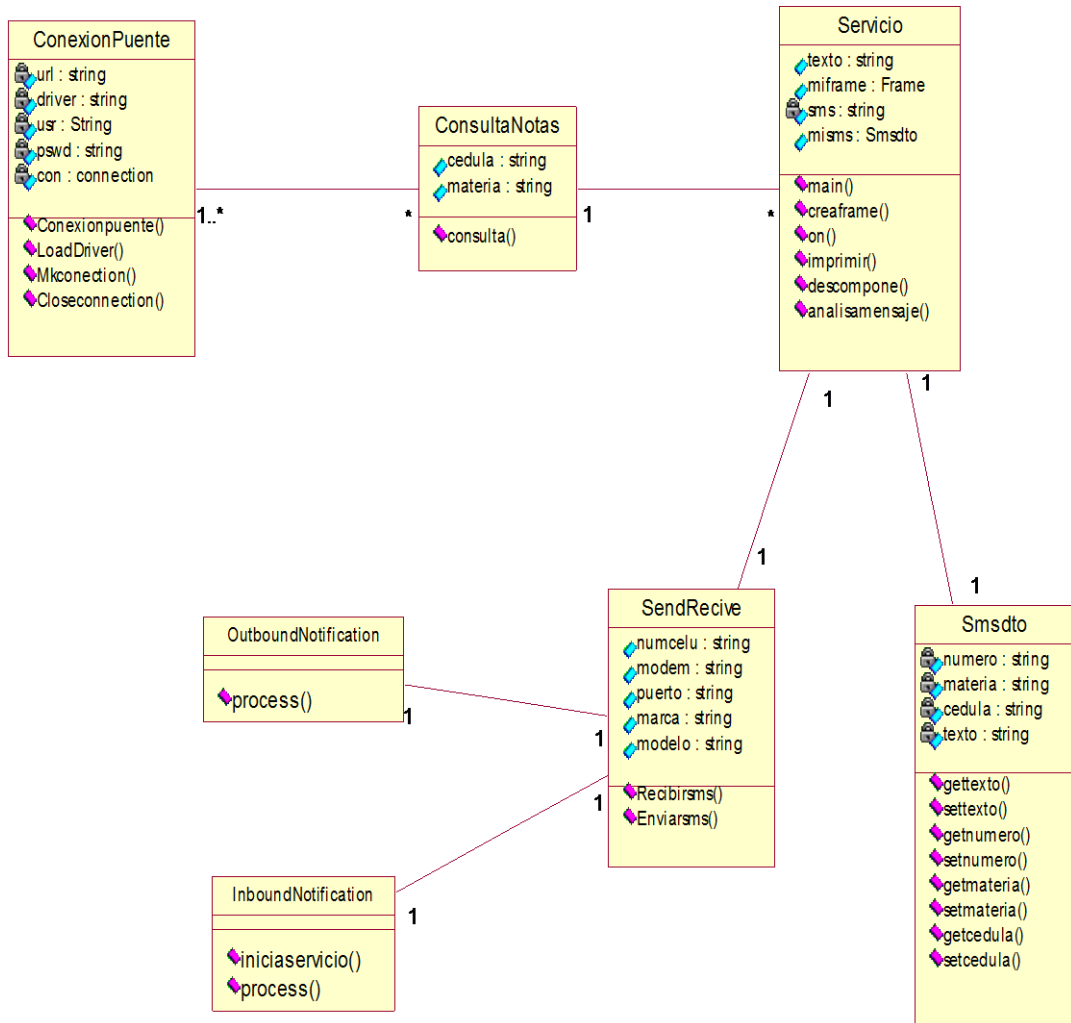


Diagrama 4.18: Diagrama de clases

4.4 IMPLEMENTACIÓN

La aplicación está dividida en 3 paquetes que contienen las diferentes clases:

- fav.tesis.sms.bdd
- fav.tesis.sms.servicio
- fav.tesis.sms.sms

El paquete fav.tesis.sms.bdd contiene las clases que hacen referencia a la conexión con la base de datos y la escritura de un archivo que contendrá un registro de la ejecución de la aplicación. Este paquete contiene las siguientes clases:

- conexionpuente
- WriteLog

En el paquete fav.tesis.sms.servicio estarán las clases que inician los servicios y realizan las llamadas a otras clases para realizar las diferentes tareas requeridas, estas clases son:

- ConsultaNotas
- Servicio

Y finalmente el paquete fav.tesis.sms.sms contiene las clases que realizan la conexión con el teléfono celular y el manejo de las bandejas del mismo y están son:

- CallNotification
- InboundNotification
- OutboundNotification
- SendRecive
- SmsDTO

La aplicación emplea librerías de libre distribución para su funcionamiento, las mismas que se listaran y describirán a continuación:

- Smslib.- es la biblioteca de un programador (librería) para enviar y recibir mensajes SMS a través de un módem GSM o teléfono móvil.
- Comm.- es una extensión standard que nos permite realizar comunicaciones con los puertos serie RS-232 y el paralelo IEEE-1284, esto nos permitirá realizar aplicaciones de comunicaciones que utilizan los

puertos de comunicaciones (tarjetas inteligentes, fax) independientes de la plataforma.

- Log4j.- es el API en java de referencia para realizar logeo. Su configuración a través de ficheros, sus propiedades permiten modificar en tiempos de ejecución las acciones y los niveles de evento a registrar, así como la salida de los mensajes generados.
- JConnect.- es una implementación en el lenguaje Java del estándar JDBC que proporciona a los desarrolladores que trabajan con Java un acceso nativo a bases de datos en entornos heterogéneos y de varios niveles.

A continuación se describirá cada clase y se mostrará el código de la misma:

ConexionPuente

En esta clase se realiza la conexión de la aplicación con la base de datos a través de un ODBC creado y empleando un puente JDBC-ODBC.

```
package fav.tesis.sms.bdd;
import java.sql.*;
public class conexionpuente {

    /*Atributos de la conexión*/
    private String url = "jdbc:odbc:";
    private String driver = "ACAD_LATA";
    private String usr = "usrweb_alumno";
    private String pswd = "usrweb";
    private Connection con;

    /*Crea el constructor, carga puente JDBC-ODBC*/
    public conexionpuente()
    {
        loadDriver();
    }
    /**
    * Carga el driver de la conexión a la base de datos
    */
    private void loadDriver()
    {
        try
```

```

    {
        //Instancia de una nueva clase para el puente
        //sun.jdbc.odbc.JdbcOdbcDriver
        //El puente sirve entre la aplicación y el driver.
        Class.forName( "sun.jdbc.odbc.JdbcOdbcDriver" );
    }
    catch(ClassNotFoundException e)
    {
        System.out.println("Error al crear el puente JDBC-ODBC");
    }
}
/**
 *Obtiene una conexión con el nombre del driver especificado
 *@param driverName Nombre del driver de la base de datos
 *@return
 */
public Connection mkConection()
{
    url = url + driver;
    System.out.println("Estableciendo conexión con " + url);
    try
    {
        //Obtenemos la conexión
        con = DriverManager.getConnection( url,usr,pswd);
    }
    catch(SQLException sqle)
    {
        System.out.println("No se pudo establecer la conexión");
        return null;
    }
    System.out.println("Conexión establecida con: " + url);
    //Regresa la conexión
    return con;
}

/* Cerrar la conexión.*/

public boolean closeConecction()
{

```



```

    try
    {
        con.close();
    }
    catch(SQLException sqle)
    {
        System.out.println("No se cerró la conexión");
        return false;
    }
    System.out.println("Conexión cerrada con éxito ");
    return true;
}

```

WriteLog

Configura un archivo en una dirección especificada y escribe en él un registro de varios mensajes de la ejecución de la aplicación.

```

package fav.tesis.sms.bdd;
import java.io.*;
public class WriteLog {

    private static PrintWriter myOutWriter = null;
    // Variable que define si se debe escribir o no
    private static boolean LOG = true;
    /**
     * Setea un file para realizar la salida
     *
     * @param pNameFile
     *         nombre del file.
     */
    public static void setOut(String pNameFile) {
        try {
            myOutWriter = new PrintWriter(new FileWriter(pNameFile,
                true));
        } catch (IOException io) {
            System.out.println("Error al intentar abrir el archivo"
                + pNameFile);
            System.exit(0);
        }
    }
}

```

```

    }
}

/**
 * Setea un OutputStream para realizar la salida que puede ser la
 * consola por ejemplo.
 *
 * @param pOut
 */

public static void setOut(OutputStream pOut) {
    myOutWriter = new PrintWriter(pOut);
}

/**
 * Logea el string que se pasa como parámetro, este proceso lo
 * realiza si la variable LOG esta en true, se le puede añadir el
 * flush en este método o utilizar a mano, según se prefiera
 *
 * @param pLine
 */

public static void write(String pLine) {
    if (LOG) {
        myOutWriter.println(pLine);
    }
}

public static void flush() {
    myOutWriter.flush();
}

/**
 * Setea la variable LOG, que indica si se debe escribir o no.
 *
 */

public static void close() {

```

```

        LOG = false;
    }

    public static void open() {
        LOG = true;
    }
}

```

ConsultaNotas

En esta clase se realiza la consulta a la base de datos con los datos obtenidos del mensaje enviado por el usuario y se estructura el mensaje con el resultado de la consulta.

```

package fav.tesis.sms.servicio;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import fav.tesis.sms.bdd.conexionpuente;

public class ConsutaNotas {

    // Realizamos la consulta de notas
    public static String consulta (String cedula, String materia)
    throws Throwable
    {
        String salida = "Error no existe estudiante o materia";
        Connection con = null;

        try {
            // Crea un objeto de la clase conexión
            conexionpuente conbdd = new conexionpuente();

            //Obtenemos la conexión
            con = conbdd.mkConection();

```

```

// Eliminamos los posibles espacios introducidos en la
// cedula
String cedulam = cedula.trim();
// Eliminamos los posibles espacios introducidos en la
// materia y convertimos a mayusculas
String materiam = materia.trim().toUpperCase();
// Realizamos la consulta

String sql = " select distinct
temp_nota_periodo.ape_est_esp " +
" ,temp_nota_periodo.nom_est_esp, materia.des_materia " +
" ,temp_nota_periodo.nota1, temp_nota_periodo.nota2 " +
" ,temp_nota_periodo.nota3" +
" from temp_nota_periodo, materia " +
" where temp_nota_periodo.cod_materia = materia.cod_materia
" + and temp_nota_periodo.ced_est_esp = '" + cedulam + "'
" + and materia.des_materia = '" + materiam + "' ";

System.out.println(sql);
Statement sta = con.createStatement();
// Ejecutamos la consulta
ResultSet rs = sta.executeQuery(sql);

// Recorremos el registro resultado de la consulta
while ( rs.next())
{
    // Concatenamos los datos resultantes de la consulta
    salida = rs.getString("APE_EST_ESP");
    salida = salida + " " + rs.getString("NOM_EST_ESP") +
    ", " + rs.getString("DES_MATERIA") ;
    salida = salida + " " + rs.getString("NOTA1") + " " +
    rs.getString("NOTA2") + " " + rs.getString("NOTA3");

}
} catch (Exception e) {
    System.out.println("Error en consulta a bdd");
    e.printStackTrace(System.out);
}
}

```

```

        finally{
            if (con != null)
                // Cerramos la conexión
                con.close();
            // Retornamos un String para enviar como SMS
            return salida;
        }
    }
}

```

Servicio

Esta clase es la que inicia la aplicación y crea una ventana e imprime en ella mensajes de la ejecución de la aplicación, también comprueba las bandejas del celular en busca de mensajes existentes y envía los mensajes encontrados para analizar.

```

package fav.tesis.sms.servicio;

import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.TextArea;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.sql.Timestamp;
import java.text.ParseException;
import java.text.SimpleDateFormat;

import com.sun.org.apache.bcel.internal.generic.RET;

import fav.tesis.sms.bdd.WriteLog;
import fav.tesis.sms.sms.SendRecive;
import fav.tesis.sms.sms.SmsDTO;

```

```

public class Servicio {

    public static void main(String[] args)
    {
        try{
            //Llamada al método On
            on();
        }catch (Exception e){
            e.printStackTrace();
        }
    }

    // Creamos una ventana para visualizar la ejecución de la
    // aplicación
    public static Frame creaFrame(){
        Frame miFrame;
        // Creamos la ventana y le damos un nombre
        miFrame = new Frame( "Consola de mensajes" );
        // Especificamos el tamaño de la ventana
        miFrame.setSize( 600 ,600 );
        // Establecemos la ventana como visible
        miFrame.setVisible( true );
        // Creamos un cuadro de texto

        TextArea miAreaTexto = new
        TextArea("",20,80,TextArea.SCROLLBARS_VERTICAL_ONLY);
        miFrame.setLayout( new FlowLayout() );
        // Añadimos el área de texto a la ventana
        miFrame.add( miAreaTexto );
        miFrame.addWindowListener( new WindowAdapter() {
            // Hacemos que al cerrar la ventana pare la ejecución de la
            // aplicación

            public void windowClosing( WindowEvent evt ) {
                System.exit( 0 );
            }
        } );
    }
}

```

```

        return miFrame;
    }

    // Escribimos mensajes en la ventana
    public static void imprimir(Frame miFrame, String texto)
    {
        try {
            TextArea miAreaTexto = (TextArea)miFrame.getComponent(0);
            miAreaTexto.append(texto + "\n");
            miFrame.setVisible( true );
            // Creamos un archivo de texto en c:/
            WriteLog.setOut("c:/logSMS.txt");
            // Escribimos mensajes en el archivo de texto
            WriteLog.write(texto);
            WriteLog.flush();
        } catch (Exception e) {

            e.printStackTrace(System.out);
        }
    }

    // Método que ejecuta la aplicación
    public static void on()
    {
        try {
            System.out.println("On");
            //Se crea la ventana
            Frame miFrame = creaFrame();

            // Crea un bucle infinito para que la aplicación
            // esté en ejecución indefinidamente
            int n = 100;
            while (n >0)
            {
                // Crea un objeto de la clase SendRecive
                SendRecive senre = new SendRecive();
                // Imprime un mensaje en la ventana
                String mensaje = senre.RecibirSms(miFrame);
                try {

```

```

        // Comprueba que el mensaje ingresado no
        //este vacio
        if (mensaje.trim().length() > 0)
        {
            // Obtiene el SMS para descomponerlo
            SmsDTO sms = Servicio.descompone(mensaje);
            // Obtiene la cedula y materia del mensaje
            // recibido
            String salio =
            ConsutaNotas.consulta(sms.getCedula(),
            sms.getMateria());
            System.out.println(salio);
            // Envía el mensaje resultado de la
            consulta
            senre.EnviaSms(sms.getNumero(), salio,
            miFrame);
        }

    } catch (Throwable e) {

        e.printStackTrace();

    }

}

} catch (Exception e) {
    e.printStackTrace();
}

}

// Descomponemos el SMS
private static SmsDTO descompone(String sms)
{
    // Cogemos el número de origen
    SmsDTO misms = new SmsDTO();
    Integer n = sms.indexOf("Originator:", 0) + 11;
    Integer end = sms.indexOf("\n", n);
    if (end > n) {
        String num = sms.substring(n, end).trim();
        // Se reemplaza los caracteres internacionales
        // por el numero 0 para enviar el mensaje al remitente
    }
}

```



```

        if ( num.length() == 12 )
        {
            num = num.replaceFirst("593","0");
        }

        misms.setNumero(num);
    }// Cogemos el texto
    n = sms.indexOf("Text:", 0) + 5;
    end = sms.indexOf("\n", n);
    if (end > n){
        misms.setTexto(sms.substring(n, end).trim());
    }
    // Enviamos el mensaje a analizar
    misms = analizaMensaje(misms);
    try {
        // Creamos una cadena de texto con la respuesta del
        //análisis
        String respuesta =
        ConsutaNotas.consulta(misms.getCedula(),
        misms.getMateria());
    } catch (Throwable e) {

        e.printStackTrace();
    }
    // Retornamos el texto del mensaje
    return misms;
}

// Analizamos el SMS
public static SmsDTO analizaMensaje(SmsDTO misms){
    if (null!=misms.getTexto() &&
    misms.getTexto().trim().length()>0)
    {
        // Especificamos el carácter separador entre
        // la cédula y la materia
        int n = misms.getTexto().indexOf(".");
        // Obtenemos la cedula
        String cedula = misms.getTexto().substring(0,n);
        // Obtenemos la materia

```

```

        String materia =
        misms.getTexto().substring(n+1,misms.getTexto().trim().
        length()).trim();
        // Almacenamos la cedula y la materia en misms
        misms.setCedula(cedula);
        misms.setMateria(materia);
    }
    // Retornamos la cedula y la materia
    return misms;
}
}

```

CallNotification

Utiliza un método propio de la librería smslib para notificar cuando existen eventos nuevos en las bandejas del celular.

```

package fav.tesis.sms.sms;

import org.smslib.ICallNotification;
/**
 * Implementa un método contenido en la librería
 * de comunicaciones smslib, que realiza una notificación
 * cuando existe un nuevo evento
 */
public class CallNotification implements ICallNotification
{
    public void process(String gatewayId, String callerId)
    {
        System.out.println(">>> Nueva llamada detectada desde el
        Gateway: " + gatewayId + " : " + callerId);
    }
}

```

InboundNotification

Utiliza un método propio de la librería smslib para notificar cuando existen eventos nuevos en la bandeja de entrada del celular.

```

package fav.tesis.sms.sms;

import java.util.LinkedList;

```

```

import org.smslib.IInboundMessageNotification;
import org.smslib.InboundMessage;
import org.smslib.Message;
import org.smslib.Service;

/**
 * Implementa un método contenido en la librería
 * de comunicaciones smslib, que realiza una notificación
 * cuando llega un nuevo mensaje a la bandeja de entrada del celular
 */
public class InboundNotification implements IInboundMessageNotification
{
    private Service srv = null;

    public void iniciaservicio(Service srv1)
    {
        this.srv = srv1;
    }

    public void process(String gatewayId, Message.MessageTypes msgType,
String memLoc, int memIndex)
    {
        LinkedList<InboundMessage> msgList;

        switch (msgType)
        {
            case INBOUND:
                System.out.println(">>> Nuevo mensaje de entrada
detectado desde el Gateway: " + gatewayId + " : "
+ memLoc + " @ " + memIndex);
                try
                {
                    // Lee...
                    msgList = new LinkedList<InboundMessage>();
                    // Se comprueba que el mensaje no haya sido
leído

```

```

        srv.readMessages(msgList,
            InboundMessage.MessageClasses.UNREAD,
            gatewayId);
        for (int i = 0; i < msgList.size(); i++)

            System.out.println(msgList.get(i).toString());
            // ...y responde.

        }
        catch (Exception e)
        {
            System.out.println("Error al leer bandeja
                de entrada");
            e.printStackTrace();
        }
        break;

    case STATUSREPORT:
        System.out.println(">>> Nuevo mensaje de reporte
            de estado detectado desde el Gateway: " +
            gatewayId + " : " + memLoc + " @ " + memIndex);
        break;

    }
}
}

```

OutboundNotification

Utiliza un método propio de la librería smslib para notificar cuando existen eventos nuevos en la bandeja de salida del celular.

```

package fav.tesis.sms.sms;

import org.smslib.IOutboundMessageNotification;
import org.smslib.OutboundMessage;

/**
 * Implementa un método contenido en la librería
 * de comunicaciones smslib, que realiza una notificación
 * cuando hay un mensaje en la bandeja de salida del celular
 */

```

```

public class OutboundNotification implements IOutboundMessageNotification
{
    public void process(String gatewayId, OutboundMessage msg)
    {
        System.out.println("Controlador de llamadas salientes desde
el Gateway: " + gatewayId);
        System.out.println(msg);
    }
}

```

SendRecive

En esta clase se realiza la conexión entre el teléfono celular y el computador y se envían y obtienen los mensajes desde y hacia las bandejas del celular.

```

package fav.tesis.sms.sms;

import java.awt.Frame;
import java.util.Date;
import java.util.LinkedList;
import org.smslib.InboundMessage;
import org.smslib.OutboundMessage;
import org.smslib.Service;
import org.smslib.gateway.ModemGateway;
import org.smslib.gateway.SerialModemGateway;
import fav.tesis.sms.servicio.Servicio;

/**
 * Métodos que interactúan con el celular y el puerto de
comunicaciones
 */
public class SendRecive {

    // Parámetros requeridos para la comunicación
    // con el teléfono celular
    final static String NUMCELU="+083285094";
    final static String MODEM="Motorola USB Modem";
    final static String PUERTO="COM4";
    final static String MARCA="Motorola";
    final static String MODELO="L6";

```

```

private ModemGateway gateway = null;
private Service srv;

//Método para recibir mensajes desde el celular (el último no
leído)
public String RecibirSms(Frame miFrame)
{
    // Encera la variable de salida
    String salida="";
    // Obtiene la hora del sistema
    Date hora = new Date(System.currentTimeMillis());
    // Imprime el mensaje de inicio de recepción más la hora en
la ventana
    Servicio.imprimir(miFrame, "Inicia Recepcion....." + hora );
    //System.out.println("Inicia Recepción....." + hora );
    // Obtiene el mensaje de la bandeja de entrada
    LinkedList<InboundMessage> msgList;
    // Llamam a los métodos de la librería smslib
    InboundNotification inboundNotification = new
InboundNotification();

    CallNotification callNotification = new CallNotification();
    try{
        try
        {
            // Crea un servicio para la conexión del
            celular
            srv = new Service();
            // Envía los parámetros de conexión
            gateway = new SerialModemGateway(MODEM,
            PUERTO, 57600, MARCA, MODELO,
            srv.getLogger());
            // Establece como activa la bandeja de
            entrada
            gateway.setInbound(true);
            // Envían las peticiones de notificación

gateway.setInboundNotification(inboundNotification);

```

```

gateway.setCallNotification(callNotification);

    // Crea el Gateway con los parametros
    establecidos
    srv.addGateway(gateway);
    // Inicia el servicio
    srv.startService();
    // Crea una lista de los mensajes de la
    bandeja de entrada
    msgList = new LinkedList<InboundMessage>();
    // Lee el ultimo mensaje no leído
    srv.readMessages(msgList,
    InboundMessage.MessageClasses.UNREAD);

    // Comprueba si existen mensajes sin leer
    if (msgList.size() > 0){
        // Carga el contenido del mensaje en
        la variable salida
        salida = msgList.get(0).toString();
    }
    else
        // Imprime un mensaje en la ventana
        Servicio.imprimir(miFrame, "No existen
        mensajes nuevos");

    // Borra los mensajes leídos
    srv.deleteMessage(msgList.get(0));
    // Para el servicio
    srv.stopService();

    // Comprueba si esta en ejecución el
    servicio
    if (gateway != null && gateway.isStarted())
    {
        // Cierra la conexión con el celular

        gateway.getModemDriver().disconnect();
        gateway = null;
    }

```

```

    }
    // en caso de error
    catch (Throwable e)
    {
        e.printStackTrace(System.out);

        // Comprueba si esta en ejecución el
        servicio
        if (gateway != null && gateway.isStarted())
        {
            // Cierra la conexión con el celular

            gateway.getModemDriver().disconnect();
            gateway = null;
            if (srv != null)
                // Para el servicio
                srv.stopService();
        }
    } finally
    {
        // Para el servicio
        srv.stopService();
        gateway = null;
    }
}
catch (Exception e)
{
    e.printStackTrace();
    // Imprime en la ventana el contenido de la variable
    salida
    Servicio.imprimir(miFrame, "Salida: " + salida);
}
//System.out.println("Finaliza Recepción.....");
// Imprime en la ventana un mensaje de finalización de recepción
Servicio.imprimir(miFrame, "Finaliza Recepcion.....");
Servicio.imprimir(miFrame, salida);
//System.out.println(salida);
return salida;

```



```

}
/** Envía un mensaje de texto a través del celular
    **/
public String EnviarSms(String numero, String mensaje, Frame
miFrame) throws Exception
{
    String enviado="N";
    OutboundMessage msg= null;
        try {
            OutboundNotification outboundNotification = new
            OutboundNotification();
            // Inicia el servicio
            srv = new Service();
            // Envió los parámetros de conexión
            gateway = new
            SerialModemGateway(MODEM, PUERTO, 57600, MARCA,
            MODELO, srv.getLogger());
            // Establece como activa la bandeja de salida
            gateway.setOutbound(true);

gateway.setOutboundNotification(outboundNotification);
            // Crea el Gateway
            srv.addGateway(gateway);
            // Inicia el servicio
            srv.startService();
            // Envía una mensaje sincronizadamente.
            if (!(numero.substring(0,1).equals("+")))
                {numero="+"+numero;}
            // Obtiene el numero y el mensaje
            msg = new OutboundMessage((numero), mensaje);
            // Establece el numero de origen
            msg.setOriginator(NUMCELU);
            // Declaramos que el mensaje no es de tipo flash
            msg.setFlashSms(false);
            // Envío el mensaje
            srv.sendMessage(msg);
            srv.queueMessage(msg, gateway.getGatewayId());
            // Comprueba que el servicio este en ejecución
            if (gateway != null && gateway.isStarted())

```

```

    {
        // Para el servicio
        gateway.getModemDriver().disconnect();
        if (srv != null)
            srv.stopService();
    }
    // Imprime mensaje en la ventana de mensaje
    enviado
    Servicio.imprimir(miFrame, "Mensaje enviado: "
    +mensaje);
    //System.out.println(" El mensaje fue enviado ");
    enviado="S";
}
catch (Throwable e)
{
    e.printStackTrace();
    //System.out.println("error" + e.getMessage());
    e.printStackTrace(System.out);
    enviado="N";
    // Comprueba si el servicio se está ejecutando
    if (gateway != null && gateway.isStarted())
    {
        // Para el servicio
        gateway.getModemDriver().disconnect();
        if (srv != null)
            srv.stopService();
        gateway = null;
    }
}
gateway = null;
return enviado;
}
}

```

SmsDTO

En esta clase se obtienen tanto el texto del mensaje como el número del usuario que lo envía, también la materia y el número de cédula que está en el mensaje.

```
package fav.tesis.sms.sms;
```

```

public class SmsDTO {

    // Parámetros del mensaje
    private String numero;
    private String materia;
    private String cedula;
    private String texto;

    // Obtiene el texto del mensaje
    public String getTexto() {
        return texto;
    }
    public void setTexto(String texto) {
        this.texto = texto;
    }
    // Obtiene el numero del remitente
    public String getNumero() {
        return numero;
    }
    public void setNumero(String numero) {
        this.numero = numero;
    }
    // Obtiene el nombre de la materia
    public String getMateria() {
        return materia;
    }
    public void setMateria(String materia) {
        this.materia = materia;
    }
    // Obtiene la cedula
    public String getCedula() {
        return cedula;
    }
    public void setCedula(String cedula) {
        this.cedula = cedula;
    }
}

```

5. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- La creación de una aplicación para consultar notas a través de mensajes de texto se pudo llevar a cabo, en el desarrollo de la misma se encontraron varios problemas de compatibilidad con la base de datos, pero finalmente se logro el objetivo.
- Los teléfonos celulares hoy en día se han convertido en una herramienta indispensable para muchas personas en sus trabajos y en su vida diaria, por este motivo han aparecido cada vez más utilidades aplicadas a dichos teléfonos, y se pueden crear un sinnúmero más de aplicaciones para sacar un mejor provecho de ésta tecnología.
- El uso de mensajes de texto (SMS) en nuestro país es empleado mayoritariamente por jóvenes, siendo éste un medio eficiente para la comunicación con este sector de la población.
- El modelamiento de datos mediante el empleo de diagramas UML ayudan a entender y posteriormente codificar de mejor manera el proyecto.
- Existen herramientas de modelamiento como “Rational Rose” que ayudan a la creación de diagramas UML.
- Java es un lenguaje de programación simple, seguro, robusto pero potente, que además posee una gran cantidad de librerías de libre distribución que facilitan en gran manera el desarrollo de tareas y una comunidad de desarrolladores que pueden ayudar ante cualquier problema por lo que lo convierten en una excelente herramienta de desarrollo.

- Eclipse es un IDE de Java que ofrece muchas facilidades a la hora de crear aplicaciones, ya que se puede adicionar varias utilidades más en forma de plugin de fácil instalación según se vaya necesitando.
- El manejo de programación orientada a objetos puede ser complicada al inicio pero permite una programación óptima con un uso eficiente de código cuando se llega a entender este paradigma.
- Existen librerías en Java que facilitan nuestro trabajo ya que realizan tareas que tomarían mucho tiempo y esfuerzo realizarlas con solo hacer una llamada a métodos contenidos en ellas, y de igual manera existen ejemplos en internet de clases que hacen uso de dichas librerías que ayudan a su mejor aplicación.

5.2 RECOMENDACIONES

- Es recomendable usar un teléfono celular de una marca que brinde soporte y drivers para proporcionar la compatibilidad con los diferentes sistemas operativos y aplicaciones.
- Listar claramente los requisitos sin omitir ninguno, lo que ayudará a que se implementen todos y se cumpla a cabalidad el objetivo de la aplicación.
- Agrupar en paquetes las clases que tengan una relación directa ayuda a una mejor organización y un mejor entendimiento de la aplicación.
- Ocupar nombres tanto de clases, métodos, variables, etc. que tengan relación con los mismos para poder comprender mejor el código de la aplicación.
- Revisar si existe compatibilidad de la aplicación con la base de datos y con el método de acceso a la base de datos para prevenir posteriores problemas.

- Buscar librerías en internet que cumplan con las tareas que se necesiten antes de programar estas tareas, dado a esto llevaría mucho tiempo y esfuerzo.
- Importar las librerías dentro del espacio de trabajo para que al momento de acceder a ellas se lo realice de una manera fácil.

ANEXO 1

GLOSARIO DE TÉRMINOS

AMPS.- El sistema telefónico móvil avanzado o AMPS es un sistema de telefonía móvil de primera generación (1G, voz analógica) desarrollado por los laboratorios Bell.

APPLET.- Un applet es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web.

ARRAY.- Una matriz es una disposición sistemática de los objetos, por lo general en filas y columnas. En concreto, puede referirse a varias cosas.

ASCII.- El código estadounidense estándar para el intercambio de información, o ASCII, se basa en un conjunto de caracteres del alfabeto latino utilizado en el idioma inglés y otras lenguas del mundo occidental.

BIT.- Un bit es una señal electrónica que puede estar encendida (1) o apagada (0). Es la unidad más pequeña de información que utiliza un ordenador. Son necesarios 8 bits para crear un byte.

BUS.- En arquitectura de computadores, el bus es un sistema digital que transfiere datos entre los componentes de un ordenador o entre ordenadores.

BYTECODE.- Es un código intermedio más abstracto que el código máquina.

CACHE.- Es un sistema especial de almacenamiento de alta velocidad.

CIFRADO.- Es una de las técnicas de codificación más simples y más usadas. Es un tipo de cifrado por sustitución en el que una letra en el texto original es reemplazada por otra letra que se encuentra tres posiciones más adelante en el alfabeto.

COMPILADOR.- Es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente que la máquina será capaz de interpretar.

CONCENTRADORES.- Un concentrador o hub es un dispositivo que permite centralizar el cableado de una red y poder ampliarla.

C++.- Es un lenguaje de programación, con mecanismos que permitan la manipulación de objetos.

CVS.- Es una aplicación informática que implementa un sistema de control de versiones: mantiene el registro de todo el trabajo y los cambios en los ficheros

(código fuente principalmente) que forman un proyecto (de programa) y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren.

FCC.- Comisión Federal de las Comunicaciones es una agencia estatal independiente de Estados Unidos, bajo responsabilidad directa del Congreso.

FRAGMENTACIÓN.- En computación, es el fenómeno de división de datos.

FRAMEWORKS.- Un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

GBPS.- Gigabit por segundo (a menudo abreviado por su sigla Gbps) es, en Telemática, la velocidad de transmisión de información. No se puede confundir con la unidad de información el Bit.

HEXADECIMAL.-Sistema numérico en base 16.

IDE.- Controla los dispositivos de almacenamiento masivo de datos, como los discos duros y WATAPI (*WAdvanced Technology Attachment Packet Interface*) y además añade dispositivos como las unidades CD-ROM.

IP.- Protocolo para la comunicación en una red a través de paquetes conmutados, es principalmente usado en Internet.

JAVA.- Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general.

KBPS.- Un kbps equivale a 1000 bits por segundo. Generalmente usado para medir velocidades de conexión o transferencias en una red.

LIBRERÍAS.- Las librerías proporcionan una interfaz abstracta para tareas que son altamente dependientes del hardware de la plataforma destino y de su sistema operativo.

MBIT/S.- Un megabit por segundo (Mb/s o también Mbit/s) es una unidad que se usa para cuantificar un caudal de datos equivalente a 1000 kilobits por segundo o 1000000 bits por segundo.

MHZ.- Equivale a un millón de hertz (hercios). Los MHz son empleados para medir las frecuencias de reloj de las CPU en las computadoras.

MMS.- Servicio de Mensajería Multimedia, que combina imágenes, sonido y texto en un mismo mensaje.

MTSO.- Teléfono Celular Oficina (MTSO), un ordenador o un interruptor, es el cerebro de un sistema celular. Cada celular tiene una o más MTSOs o interruptores.

METADATA.- Literalmente «sobre datos», son datos que describen otros datos.

MÓDEM.- Un módem es un dispositivo que sirve para enviar una señal llamada portadora mediante otra señal de entrada llamada moduladora.

MULTIPLEXADO.- Proceso consistente en recibir mensajes de diferentes fuentes y enviarlas a un destino común. A la inversa, la técnica de multiplexado permite enviar a puntos de destino diversos datos que proceden de una fuente común.

OCTETOS.- Conjunto significativo de ocho bits que representan un carácter, por ejemplo la letra "a", en un sistema informático.

OHMS.- Establece que "la intensidad de la corriente eléctrica que circula por un conductor eléctrico es directamente proporcional a la diferencia de potencial aplicada e inversamente proporcional a la resistencia del mismo".

PATH.- Camino que toma el software para acceder a los datos contenidos en una unidad de almacenamiento.

PERIFÉRICOS.- En informática, se denominan periféricos a los aparatos o dispositivos auxiliares e independientes conectados a la unidad central de procesamiento de una computadora.

PROTOCOLOS.- Ciertas reglas establecidas para las ceremonias oficiales o trato social.

PRIVATE.- Los elementos declarados como Private son accesibles sólo desde la misma clase donde fueron definidos.

PROTECTED.- Son elementos declarados como Protected son accesibles desde la misma clase donde fueron definidos y en sus subclases.

PUBLIC.- Los elementos declarados como Public son accesibles tanto desde fuera como desde dentro de la clase.

PLUG AND PLAY.- Plug-and-play es la tecnología que permite a un dispositivo informático ser conectado a un ordenador sin tener que configurar sus controladores.

PLUGIN.- Programa que puede anexarse a otro para aumentar sus funcionalidades (generalmente sin afectar otras funciones ni afectar la aplicación principal).

RDSI.- Red de una telefónica con un ancho de banda desde los 64 kbps, que es similar a una red telefónica de voz, pero no es analógico, sino que digital.

ROAMING.- Se refiere a la capacidad de cambiar de un área de cobertura a otra sin interrupción en el servicio o pérdida en conectividad.

SDK.- Es generalmente un conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema concreto.

SIM CARD.- A pesar de este nombre complicado, nos estamos refiriendo a un dispositivo en el que se puede guardar toda la información y los datos que los usuarios tenemos en nuestra tarjeta SIM.

POLIMORFISMO.- Se refiere a la posibilidad de definir clases diferentes que tienen métodos o atributos denominados de forma idéntica, pero que se comportan de manera distinta.

RUNTIME.- Una sola instalación de un programa de software determinado o una pieza de software en un único ordenador (CPU) o un único servidor.

TCP/IP.- Es la base del Internet que sirve para enlazar computadoras que utilizan diferentes sistemas operativos.

TDD.- Usa el mismo canal de frecuencia para enviar y recibir señales sobre downlink y uplink, lo cual le da una mejor eficiencia.

THREAD.- Se puede considerar como la agrupación de un trozo de programa junto con el conjunto de registros del procesador que utiliza y una pila de máquina.

UMTS.- Sistema Universal de Telecomunicaciones Móviles es una de las tecnologías usadas por los móviles de tercera generación sucesora de GSM.

UML.- Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.

UNIX.- Sistema operativo utilizados y con más futuro debido a que son muchos organismos oficiales y particulares los que defienden su utilización, así como

muchas firmas de fabricación y comercialización de computadoras que lo incorporan en sus productos.

WAP.- Protocolo de aplicaciones inalámbricas, un estándar seguro que permite que los usuarios accedan a información de forma instantánea a través de dispositivos inalámbricos.

WIDGET.- Es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de *widgets* o Widget Engine.

WIMAX.- Es una norma de transmisión de datos usando ondas de radio.

ANEXO 2

IEEE830-1998.

Véase en el CD adjunto a este documento.

ANEXO 3

Aprendiendo UML en 24 horas.

Véase en el CD adjunto a este documento.

ANEXO 4

Manual Rational Rose 2006-2007.

Véase en el CD adjunto a este documento.

ANEXO 5

Manual de Java

Véase en el CD adjunto a este documento.

ANEXO 6

Manual de Eclipse

Véase en el CD adjunto a este documento.

ANEXO 7

Instaladores de Eclipse

Véase en el CD adjunto a este documento.

ANEXO 8

Drivers Motorola L6

Véase en el CD adjunto a este documento.

ANEXO 9

Manual de Instalación

Véase en el CD adjunto a este documento.

ANEXO 10

Manual de usuario

Véase en el CD adjunto a este documento.

REFERENCIAS BIBLIOGRÁFICAS

- http://es.wikipedia.org/wiki/Sistema_Telefónico_Móvil_Avanzado
DESCRIPCIÓN Y CARACTERÍSTICAS DEL SISTEMA AMPS
- http://www.mundocontact.com/enlinea_detalle.php?recordID=9416
ESTÁNDARES EN TELEFONÍA CELULAR
- <http://es.wikipedia.org/wiki/CDMA2000>
INFORMACIÓN DEL ESTÁNDAR CDMA2000
- <http://www.yucatan.com.mx/especiales/celular/historia>
EVOLUCIÓN E HISTORIA DE LA TELEFONÍA CELULAR
- <http://www.revistaciencias.com/publicaciones/EpZyEkkuVlwxnCdrUU.php>
HISTORIA Y GENERACIONES DE TELEFONÍA CELULAR
- http://es.wikipedia.org/wiki/Acceso_m%C3%BAltiple_por_divisi%C3%B3n_de_frecuencia
ACCESO MÚLTIPLE POR DIVISIÓN DE FRECUENCIA
- <http://www.ordenadores-y-portatiles.com/informacion-sms.html>
COMO FUNCIONA UN SMS
- http://es.wikipedia.org/w/index.php?title=Historia_del_SMS&action=edit§ion=1
HISTORIA DE UN SMS
- <http://www.lge.com/common/index.jsp>
LG ELECTRONICS
- http://translate.google.com/translate?hl=es&langpair=en%7Ces&u=http://en.wikipedia.org/wiki/Text_messaging&prev=/translate_s%3Fhl%3Des%26q%3Dsms%2Bformato%2Btexto%26tq%3Dsms%2Btext%2Bformat%26sl%3Des%26tl%3Den
MENSAJES DE TEXTO
- http://es.wikipedia.org/w/index.php?title=Red_de_celdas
RED DE CELDAS
- http://es.wikipedia.org/w/index.php?title=Servicio_de_mensajes_cortos
SERVICIOS DE MENSAJES CORTOS

- <http://www.openbsd.org.mx/2008/09/dispositivos-electronicos/sms-en-formato-pdu-2/>
SMS EN FORMATO PDU
- <http://www.xataka.com/moviles/sms-historia-y-aplicaciones-i#blog-comments>
SMS HISTORIA Y APLICACIÓN
- <http://es.wikipedia.org/wiki/TDMA>
TDMA
- http://www.mexicoglobal.net/monografias/tecnologia/composicion_tecnologica_de_celulares.asp
COMPOSICIÓN TECNOLÓGICA DE UN TELÉFONO CELULAR
- http://es.wikipedia.org/wiki/Telef%C3%B3nica_M%C3%B3viles_Ecuador
TELEFÓNICA MÓVILES ECUADOR
- <http://es.wikipedia.org/wiki/UMTS#Arquitectura>
UNIVERSAL MOBILE TELECOMMUNICATIONS SYSTEM
- http://www.patagoniasystems.com.ar/ventajas_sms.php
VENTAJAS DE LOS SMS
- [http://es.wikipedia.org/wiki/PS/2_\(puerto\)](http://es.wikipedia.org/wiki/PS/2_(puerto))
PS/2(PUERTO)
- http://es.wikipedia.org/wiki/Bluetooth_%28especificaci%C3%B3n%29
BLUETOOTH
- http://es.wikipedia.org/w/index.php?title=Enlace_infrarrojo
ENLACE INFRARROJO
- <http://es.wikipedia.org/w/index.php?title=M%C3%B3dem>
MÓDEM
- http://es.wikipedia.org/wiki/Puerto_serie
PUERTO SERIE
- http://es.wikipedia.org/w/index.php?title=Universal_Serial_Bus
USB

- <http://www.alegsa.com.ar/Notas/133.php>
CARACTERÍSTICAS DE WINDOWS VISTA
- http://es.wikipedia.org/w/index.php?title=Caso_de_uso
CASO DE USO
- http://es.wikipedia.org/wiki/Caso_de_uso
DEFINICIÓN DE CASOS DE USO
- http://es.wikipedia.org/wiki/Diagrama_de_clases
DIAGRAMA DE CLASES
- http://es.wikipedia.org/wiki/Diagrama_de_secuencia
DIAGRAMA DE SECUENCIA
- <http://www.zona-net.com/telefonos/motorola-phone-tools>
DRIVERS PARA MOTOROLA L6
- [http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software))
ECLIPSE
- http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado
LENGUAJE UNIFICADO DE MODELADO
- <http://www.alegsa.com.ar/Notas/134.php>
REQUERIMIENTOS DE WINDOWS VISTA
- http://www.mundotutoriales.com/tutoriales_java-mdpal15793.htm
TUTORIAL DE JAVA
- http://es.wikipedia.org/wiki/Windows_Vista
WINDOWS VISTA