APLICACIÓN DE SOFTWARE EN UN DISPOSITIVO MÓVIL CELULAR PARA LA SEGURIDAD DE UN AUTOMÓVIL MEDIANTE COMUNICACIÓN BLUETOOTH CON SISTEMA DE CIFRADO

RESFA MARCELA GARCÌA PÉREZ
HECTOR ANDRES PARRA CUBIDES

FUNDACIÓN UNIVERSITARIA SAN MARTÍN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA Y TELECOMUNICACIONES
BOGOTÁ, COLOMBIA
2009, JULIO

APLICACIÓN DE SOFTWARE EN UN DISPOSITIVO MÓVIL CELULAR PARA LA SEGURIDAD DE UN AUTOMÓVIL MEDIANTE COMUNICACIÓN BLUETOOTH CON SISTEMA DE CIFRADO

RESFA MARCELA GARCÍA PÉREZ
Código: 021061
marce_g_p@hotmail.com

HECTOR ANDRES PARRA CUBIDES
Código: 031073
anpacu84@hotmail.com

MONOGRAFÍA DE GRADO

ASESOR TÉCNICO
ING. IVÁN LADINO

FUNDACIÓN UNIVERSITARIA SAN MARTÍN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA Y TELECOMUNICACIONES
BOGOTÁ, COLOMBIA
2009, JULIO

Nota de Aceptación:

_____

ING. IVÁN LADINO
Asesor

_____

ING. JORGE ARÉVALO
Jurado

_____

ING. FREUD ROMERO
Jurado

Bogotá, Julio, 2009.

Agradezco Especialmente a mis padres Orlando García y Resfa Pérez, quienes con su esfuerzo y amor me apoyaron constantemente para culminar con éxito mis estudios como Ingeniera, también agradezco a Juan Camilo Velandia por su apoyo incondicional, familia y amigos que siempre estuvieron con migo.

Marcela García Pérez.

Agradezco profundamente a mis padres por todo el esfuerzo que hicieron por darme la oportunidad de estudiar esta carrera y apoyarme en todos los momentos difíciles que tuve durante este periodo de formación. De igual forma agradezco el apoyo de mi hermano y por el ejemplo que me dio, que fue indispensable para poder alcanzar tan anhelado logro.

Así mismo agradezco a toda mi familia que siempre confiaron en mí y a Ana Yamile Silva que fue un apoyo muy importante en la culminación de mi carrera.

Hector Andrés Parra Cubides

AGRADECIMIENTOS

Queremos manifestar nuestros más sinceros agradecimientos a aquellas personas que de alguna forma nos ayudaron en el desarrollo del proyecto de grado, en especial a nuestro asesor Ingeniero Iván Ladino, por su ayuda y orientación en cada uno de los procesos.

Al Ingeniero Jorge Arévalo y Fabián Giraldo, por sus valiosas opiniones, comentarios y sugerencias.

A nuestros amigos, docentes y familiares.

CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ANEXOS

1. RESUMEN

En este proyecto de grado se propone un software en Java, desarrollado con UML para visualizar, especificar, construir y documentar los métodos y procesos elaborados, este se instaló en un celular para el control y monitoreo de un automóvil:

- Estado de las puertas (cerradas o abiertas)
- Disparo de alarma
- Bloqueo y desbloqueo central del automóvil

Todo esto con el desarrollo de un hardware que se instala en el automóvil, permitiendo hacer control de la parte eléctrica.

Para crear una mayor seguridad en la comunicación, se desarrolló un algoritmo de cifrado simétrico por bloques llamado IDEA, que maneja una clave de cifrado de 128 bits, 52 subclaves de cifrado y 52 subclaves de descifrado.

2. INTRODUCCIÓN

En consecuencia a la gran inseguridad que se presenta actualmente con los automóviles, se ha visto en la necesidad de desarrollar tecnología que permita una mayor seguridad, es por eso que se decidió hacer de proyecto de grado el desarrollo de software en un celular con interfaz gráfica, el cual se comunica con el sistema de seguridad de los automóviles, por medio de tecnología Bluetooth y permite tener el control y monitoreo del automóvil en cuanto a seguridad se refiere.

El desarrollo de un sistema de seguridad para un automóvil surge del interés de crear nuevas aplicaciones en dispositivos móviles que usen la tecnología Bluetooth y así poder controlar objetos remotos a través de estos.

Este proyecto es una continuación del proyecto de grado realizado el segundo semestre del 2007 por Omar Ancizar Niño Ramos y Camilo Armando Sánchez Guzmán, que consistía en un sistema de control domótico mediante un dispositivo móvil celular con tecnología Bluetooth.

Teniendo como base el proyecto de grado de nuestros compañeros, se realizó un trabajo sobre la tecnología Bluetooth con las siguientes diferencias, como se observa en la tabla 1:

Proyecto de grado control domótico mediante un dispositivo móvil celular con tecnología Bluetooth	Proyecto de grado aplicación de software en dispositivo móvil celular para la seguridad de un automóvil
Comunicación Api	Comunicación Api Exportable
	Desarrollo de software basado en modelamiento UML
Envío de comandos sin información	Envió de comandos con confirmación
	Monitoreo del Automóvil

Tabla 1. Diferencias entre proyectos de grado.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Diseño e implementación del software y el hardware para el monitoreo y control del sistema de alarma de un automóvil a través del uso de un teléfono celular por comunicación Bluetooth, con seguridad por medio de cifrado.

3.2 OBJETIVOS ESPECÍFICOS

- Implementar el software de aplicación en el dispositivo celular a partir del modelamiento UML, desde los diagramas de casos de uso hasta sus componentes.
- Implementar el algoritmo de cifrado simétrico para la comunicación entre el dispositivo celular y el dispositivo de control que reside en el automóvil.
- Diseñar e implementar el generador de claves pseudoaleatorias.
- Diseñar e implementar la GUI (interfaz gráfica) en el celular para el monitoreo y control del automóvil.
- Implementar el hardware para el control y monitoreo del estado de las puertas (cerradas o abiertas), disparo de alarma así como del bloqueo y desbloqueo central del automóvil.
- Definir el dispositivo celular y el microcontrolador que mejor desempeño tenga para la realización de las aplicaciones de este proyecto.

4. MARCO REFERENCIAL

4.1 ANTECEDENTES

4.1.1 CONTROL DOMÓTICO MEDIANTE UN DISPOSITIVO MÓVIL CELULAR CON TECNOLOGÍA BLUETOOTH [PDG2007]

El proyecto de grado realizado por OMAR ANCÍZAR NIÑO RAMOS y CAMILO ARMANDO SANCHEZ GUZMAN, será tomado como punto de partida para nuestro desarrollo debido a que la finalidad de este proyecto fue el desarrollo de una aplicación en J2ME instalada en un celular que permitía a un receptor activar o no las luces de una casa como sala, comedor y habitación, también el abrir o cerrar una puerta todo esto fue posible por medio de la tecnología Bluetooth.

Consiste básicamente en programar un dispositivo móvil celular mediante J2ME, ya que esta es la plataforma de JAVA para dispositivos móviles, esta permite crear una aplicación o programa que le permite al usuario seleccionar la activación o no de las luces, o tener acceso a su vivienda sin necesidad de una llave que permita abrir la puerta; solo basta hacer la selección desde el celular. Una vez hecha esta selección el dispositivo se comunica de forma inalámbrica y envía la opción deseada a los módulos de Bluetooth, los cuales reciben la información proveniente del dispositivo móvil, procesando la información y controlando el dispositivo seleccionado mediante un reducido hardware que permite la interacción entre este y el interruptor de la luz o el control de acceso.

4.1.2 ALARMA DE AUTOMOVIL [FRVM2008]

La utilización de un transmisor y un receptor de radio frecuencia ha sido por mucho tiempo el modo más utilizado como base de las alarmas para automóviles, como se observa en la figura 1.

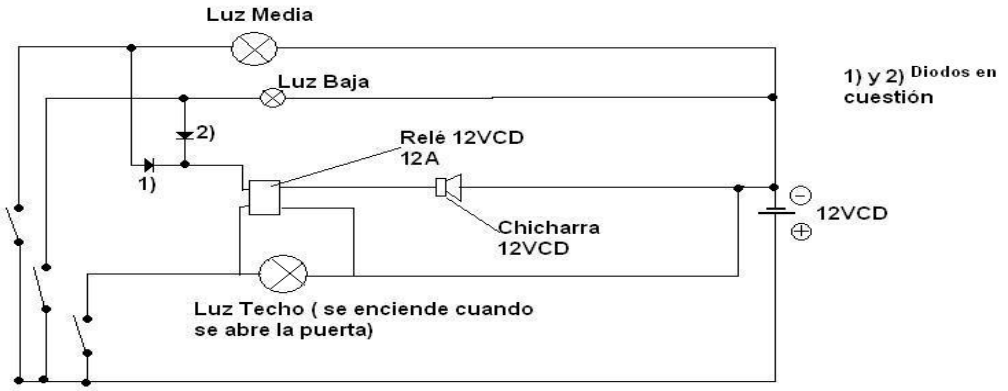


Figura 1. Esquema de conexión clásico. [YOP2009]

En la figura 2 se observa que el transmisor envía una señal al sistema de seguridad del auto para la activación o desactivación de este. Paso siguiente el receptor monitorea cada uno de los sensores disponibles tales como interruptores (ubicados en las puertas), infrarrojos (para detectar si hay alguien adentro), etc. Si alguno de estos sensores es activado el sistema produce un sonido continuo por medio de una sirena hasta el momento que el transmisor envía la señal de desactivación.

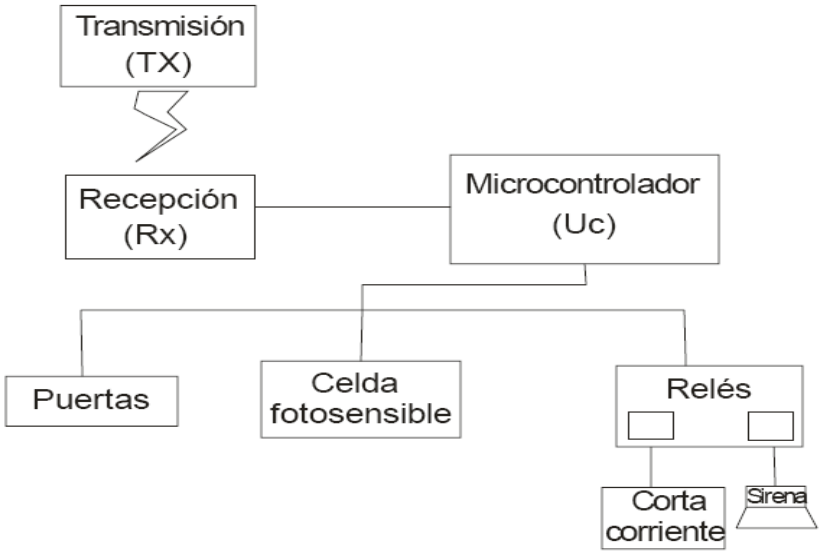


Figura 2.Esquema de funcionamiento de una Alarma convencional [FRVM2008]

4.1.3 IMPLEMENTACIÓN DEL ALGORITMO CRIPTOGRÁFICO IDEA EN VIRTEX USANDO JBITS.

Iván González y Francisco Gómez de la Universidad Autónoma de Madrid, y la Universidad Rey Juan Carlos, realizaron una implementación que fue desarrollada con JBits del algoritmo criptográfico IDEA, utilizando una FPGA de la familia Virtex por correspondiente a un sistema de mayor capacidad de cómputo que un micro controlador y un celular.

Este trabajo fue financiado por el proyecto TIC2001-2688-C03-03 del Ministerio de Ciencia e Innovación del Gobierno de España. [AIU2008]

4.2 MARCO TEÓRICO

4.2.1 BLUETOOTH [DCI2008]

Bluetooth es una tecnología de radio de corto alcance, que permite conectarse de forma inalámbrica con otros dispositivos electrónicos. Sus características principales son:

- Bajo precio.
- Mínimo consumo.
- Tamaño reducido.
- Cuenta con un dispositivo que funciona como maestro, y otro como esclavo, pudiendo haber hasta 8 dispositivos comunicándose simultáneamente, formando una Piconet (red de dispositivos informáticos que se conectan utilizando Bluetooth.)

4.2.1.1 ARQUITECTURA [SBT2008]

La arquitectura Bluetooth se organiza en "Piconets" como se muestra en la figura 3, formadas por dos o más dispositivos compartiendo un canal donde uno de los terminales actúa como el "maestro" de la Piconet, mientras que el resto actúan como esclavos.

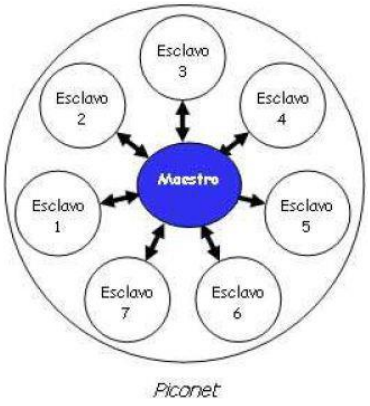


Figura 3. Piconet [SBT2008]

Bluetooth se basa en dos cosas, una tecnología de frecuencia de radio que opera en la banda de (2,402 GHz a 2,480 GHz) libre para ISM (banda de frecuencia industrial, científica y médica) que cuenta con un gran número de portadoras de espectro ensanchado por salto de frecuencia (FHSS) y el protocolo software que

le permite transmitir datos a otros dispositivos, que utiliza una modulación de frecuencia binaria.

La tasa de transferencia de símbolos es de 1 MS/s (mega símbolos por segundo), que admite una velocidad de transmisión de 1 Megabit por segundo (Mbps) en el modo de transferencia básica y una velocidad de transmisión de área total de 2 a 3 Mbps en el modo de transferencia de datos mejorada. Esta tecnología ha sido diseñada para trabajar en un ambiente llamado multi-usuario, que consiste en crear dos tipos de configuraciones que permite conformar las redes y subredes.

Existe una estructura un poco más compleja a la que se denomina una Scatternet como se observa en la figura 4, permitiendo así comunicación e intercambio de datos en configuraciones flexibles según las necesidades, como lo muestra la figura cada piconet es independiente de las demás aunque se encuentren en la misma scatternet.

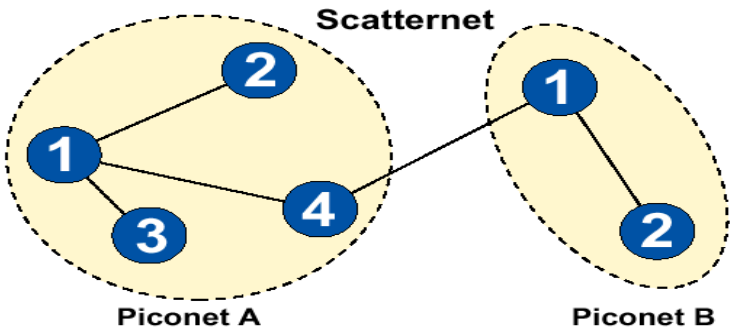


Figura 4. Estructura Scatternet [ULX2008]

4.2.1.2 CLASIFICACIÓN BLUETOOTH [DIM2008]

Existen equipos Bluetooth clase 1, 2 y 3. Las diferencias existentes en las clases, sólo afectan al alcance de la comunicación inalámbrica:

- Los dispositivos clase 1 llegan a 100 metros.
- Los dispositivos de clase 2 lo hacen a 10 metros.
- Los dispositivos de clase 3 poseen apenas un metro de alcance.

Esta clasificación de los dispositivos Bluetooth únicamente hace una referencia de la potencia de transmisión del dispositivo, siendo totalmente compatibles los dispositivos de una clase con los de la otra, como muestra la tabla 2.

Tipo	De nivel de potencia	Alcance
Dispositivos de la clase 3	100mW	Hasta 100 metros
Dispositivos de la clase 2	10mW	Hasta 10 metros
Dispositivos de la clase 1	1mW	0.1-10 metros

Tabla 2.Clasificación Bluetooth [DCI2008]

Todas las transmisiones de radiofrecuencia, dependen de:

• Obstáculos y medios físicos por donde se propaguen las ondas (paredes, muebles, gente, otros equipos).

• Interferencias con otros dispositivos inalámbricos.

• Nivel de batería de un dispositivo móvil.

Figura 5.Conexiones Bluetooth [DIM2008]

4.2.1.3 PROTOCOLOS [PROB2009]

El stack o pila de protocolos Bluetooth observados en la figura 6, se basan en el modelo de referencia OSI para la interconexión de sistemas abiertos. Las funciones de red son divididas en un sistema de niveles dentro de su arquitectura de niveles.

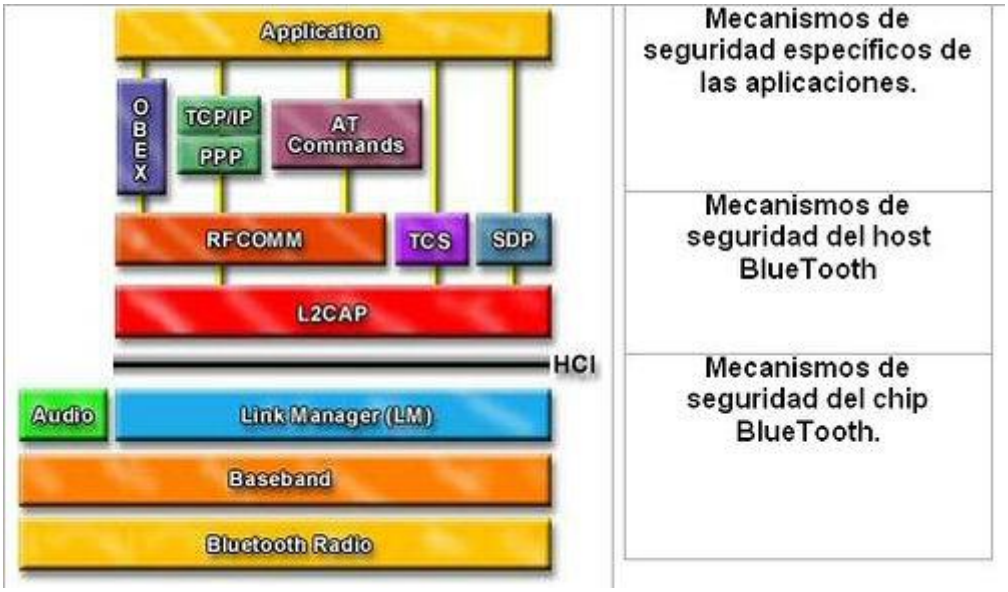


Figura 6. Pila de protocolos de Bluetooth [BEN2009]

La pila está constituida por dos clases de protocolos. Una primera clase llamada de protocolos específicos que implementa los protocolos propios de Bluetooth. Y una segunda clase formada por el conjunto de protocolos adoptados de otras especificaciones.
La pila de protocolos se puede dividir en cuatro capas lógicas:

- Núcleo de Bluetooth: Radio, Banda Base, LMP (Protocolo de gestión de enlace), L2CAP (Logical Link Control and Adaptation Protocol).
- SDP (protocolo de descubrimiento de dispositivos).
- Sustitución de cable: RFCOMM (Radio Frequency Communication).
- Protocolos adoptados: OBEX (OBject EXchange, intercambio de datos).
- TCP (protocolo de control de transmisión), IP (protocolo Internet).

4.2.1.4 PROTOCOLO UTILIZADO [PROB2009]

RFCOMM fue el protocolo utilizado, ya que soporta las configuraciones de módulos que se comunican vía Bluetooth y que proporcionan una interfaz de red, involucrando siempre a dos aplicaciones que se ejecutan en dos dispositivos distintos (los extremos de la comunicación). Entre ellos existe un segmento que los comunica, pretendiendo cubrir aquellas aplicaciones que utilizan los puertos serie de las máquinas donde se ejecutan. El segmento de comunicación es un enlace Bluetooth desde un dispositivo al otro (conexión directa).

Su propósito es emular un puerto serial. El protocolo cubre aplicaciones que usan puertos seriales de la clase que usan los PC. Así, el RFCOMM emula un control RS232 y señales de datos sobre la capa de banda base.

También Proporciona las capacidades del transporte para los servicios superiores.

4.2.1.5 JSR 82 [JAPI2002]

El objetivo de ésta especificación era definir un API (interfaz de programación de aplicaciones) estándar abierto, no propietario que pudiera ser usado en todos los dispositivos que implementen J2ME. Por consiguiente fue diseñado usando los APIs J2ME y el entorno de trabajo CLDC/MIDP en J2ME.

El API intenta ofrecer las siguientes características:

• Registro de servicios.

• Descubrimiento de dispositivos y servicios.

• Establecer conexiones con los protocolos: RFCOMM, L2CAP, OBEX

• Usar dichas conexiones para mandar y recibir datos (las comunicaciones de voz no están soportadas).

• Manejar y controlar las conexiones de comunicación.

• Ofrecer seguridad a dichas actividades.

Se divide en dos paquetes: javax.bluetooth y javax.obex. El primer paquete provee la funcionalidad para la realización de búsquedas de dispositivos, búsquedas de servicios y comunicación mediante flujos de datos. Por otro lado el paquete javax.obex permite la comunicación mediante el protocolo OBEX, se trata de un protocolo de alto nivel muy similar a HTTP (Protocolo de Transferencia de Hipertexto).

4.2.2 LENGUAJE DE MODELADO UNIFICADO (UML) [DISE2000]

UML es un lenguaje estándar que sirve para modelar un software, puede utilizarse para visualizar, especificar, construir y documentar todos los elementos que componen un sistema.

Nos ayuda a comprender grandes sistemas mediante gráficos y textos que permiten que personas que no estuvieron durante su desarrollo puedan interpretarlos.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones en:

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.

- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.

- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.

- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Existen una serie elementos que dan a UML características de completitud y de no ambigüedad, observados en la figura 7. El primer elemento son los bloques de construcción que se dividen en elementos, relaciones y diagramas. El segundo elemento son las reglas semánticas

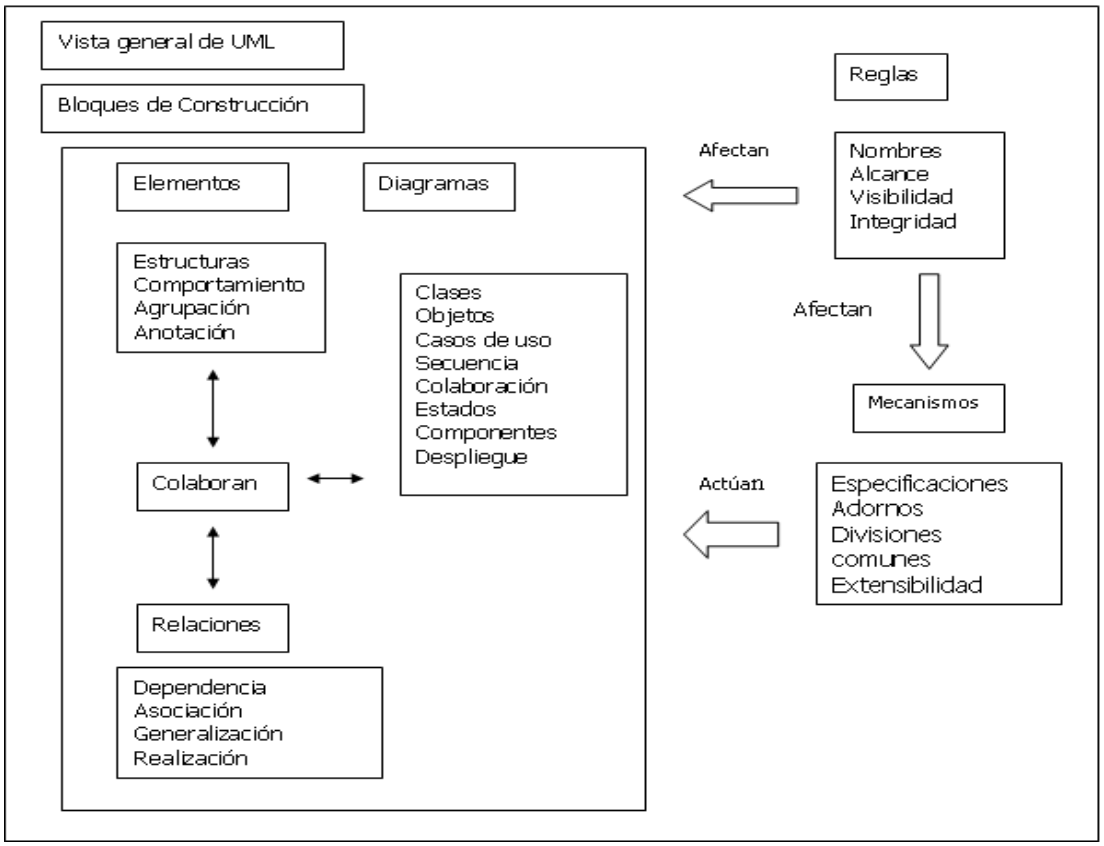


Figura 7. Elementos UML. [DISE2000]

UML es además un método formal de modelado. Esto aporta las siguientes ventajas:

• Mayor rigor en la especificación.

• Permite realizar una verificación y validación del modelo realizado.

• Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto.

4.2.2.1 ARQUITECTURA DE UML

La base de una buena arquitectura es no centrarse en la estructura y en el comportamiento si no que se expanda en la captación de diferentes temas como funcionalidad, rendimiento, capacidad de adaptación, reutilización, capacidad para ser comprendida y restricciones estéticas.

La arquitectura UML esta soportada en cinco elementos denominados vistas, observados en la figura 8:

- Vista de casos de uso: Describe el comportamiento del sistema.
- Vista de diseño: Clases, interfaces y colaboraciones que forman el vocabulario del problema y de la solución.
- Vista de procesos: Comprende procesos de sincronización del sistema.
- Vista de implementación: Componentes y archivos que son utilizados para hacer disponible el sistema físico.
- Vista de despliegue: Contiene los nodos que forman la topología hardware sobre la que se ejecuta el sistema

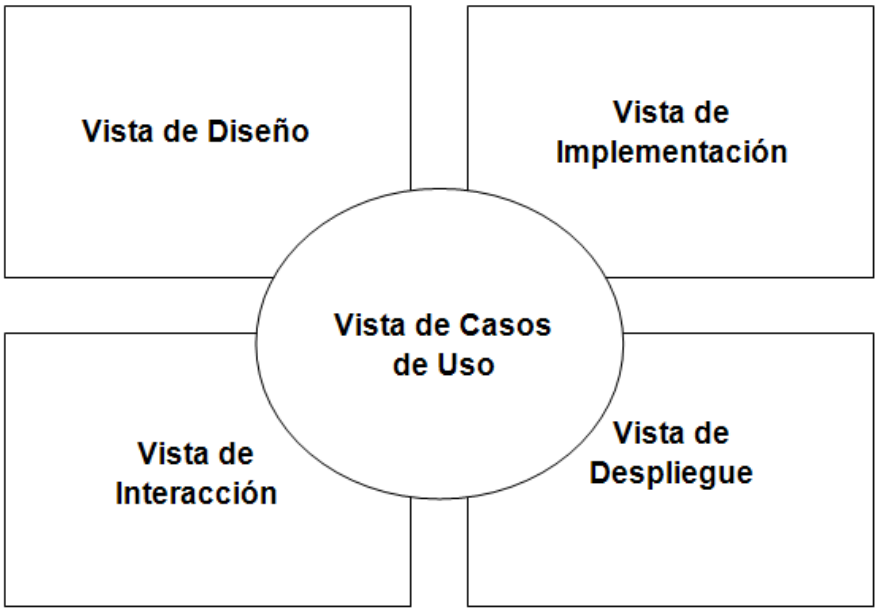


Figura 8. Modelado de la arquitectura de un sistema. [DISE2000]

4.2.2.2 DIAGRAMAS UML

Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones. Ofrece una vista del sistema a modelar. Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas. UML incluye los siguientes diagramas:

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagrama de objetos.
- Diagrama de secuencia.
- Diagrama de componentes.

Diagrama de clases: Muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones, cubren la vista de diseño estático de un sistema, incluyen clases activas cubren la vista de procesos estáticos de un sistema.

Diagrama de objetos: Muestra un conjunto de objetos y sus relaciones representan instantáneas de instancias de los elementos encontrados en los diagramas de clase. Cubren la vista de diseño y proceso estático de un sistema.

Diagrama de casos de uso: Muestra un conjunto de casos de uso y actores con sus relaciones, encargándose de cubrir la vista de casos de uso estática de un sistema.
Estos diagramas son especialmente importantes en el modelado y organización del comportamiento de un sistema.

Diagrama de secuencia: Es un diagrama de interacciones que resalta el orden temporal de los mensajes.

Diagrama de componentes: Muestra la organización y las dependencias entre un conjunto de componentes, cubren la vista de implementación estática, se relacionan con diagramas de clase en que un componente se corresponde con una o más clases, interfaces o colaboraciones.

4.2.3 CIFRADO DE DATOS.

4.2.3.1 ALGORITMO SIMÉTRICO [EYE2008]

El cifrado simétrico consiste en el uso de una clave que es conocida tanto por el emisor como por el receptor, se observa en la figura 9. El emisor genera el mensaje cifrado utilizando un algoritmo de cifrado simétrico así como la clave, para luego transmitir el mensaje cifrado al receptor. El receptor aplica la misma clave y el algoritmo inverso para obtener nuevamente el mensaje original. Este método garantiza confidencialidad y autentificación.

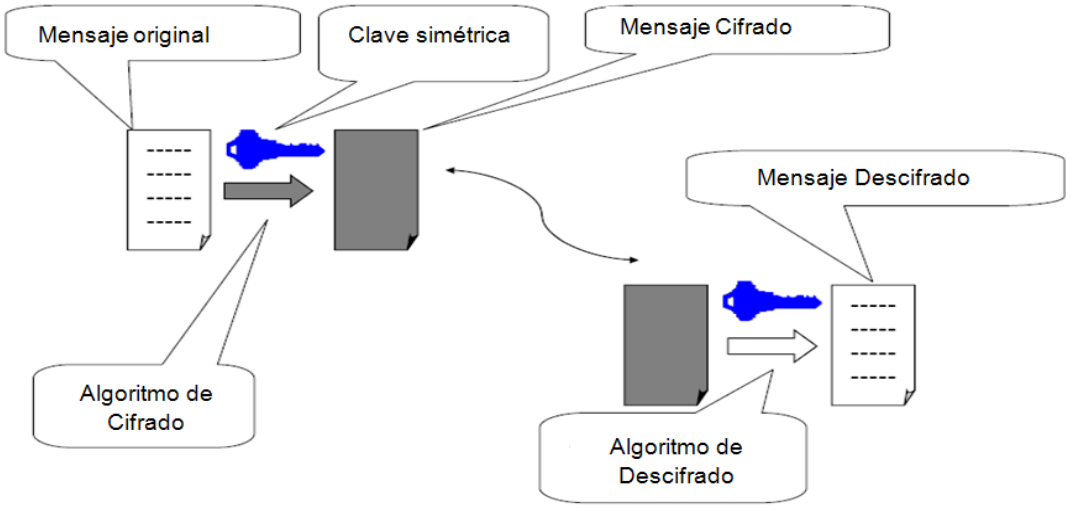


Figura 9. Esquema Simétrico. [CCC2008]

Los cifradores simétricos presentan una serie de ventajas tales como:

- Las claves para cifradores simétricos son relativamente cortas.
- Uso de operaciones matemáticas simples
- Toda la seguridad está basada en la privacidad de esta clave
- Se caracteriza por usar la misma clave para encriptar y desencriptar.
- Pueden ser empleados como primitivas para construir varios mecanismos criptográficos incluyendo generadores de números pseudoaleatorios.
- Pueden ser diseñados para tener altas tasas de transmisión de datos.

4.2.3.2 ALGORITMO ASIMÉTRICO [EYE2008]

Los algoritmos de cifrado asimétrico que se observa en la figura 10, se basan en el uso de dos claves, una pública y otra privada, de tal forma que lo que una de ellas cifra, sólo puede descifrarlo la otra, y viceversa. Tanto el emisor como el receptor poseen dos claves: una privada (conocida sólo por él) y una pública (conocida por cualquiera), de manera que no es necesario que el emisor y el receptor intercambien claves secretas. Además, sólo se necesitan un par de claves privada/pública por persona.

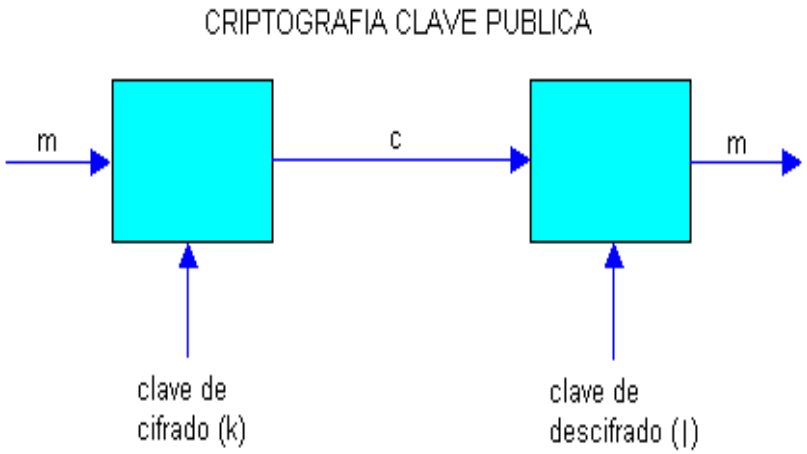


Figura 10. Esquema Asimétrico. [EYE2008]

- Emplean longitudes de clave mucho mayores que los simétricos.

- Además, la complejidad de cálculo que comportan los hace más lentos que los algoritmos de cifrado simétricos.

- Por ello, los métodos asimétricos se emplean para intercambiar la clave de sesión mientras que los simétricos para el intercambio de información dentro de una sesión.

4.2.4 J2ME

J2ME (Java 2 Plataform, Micro Edition), es la versión de Java orientada a los dispositivos móviles, como se observa en la figura 11, se ha creado para la programación de dispositivos inalámbricos pequeños como teléfonos celulares, debido a sus capacidades computacionales y gráficas muy reducidas.

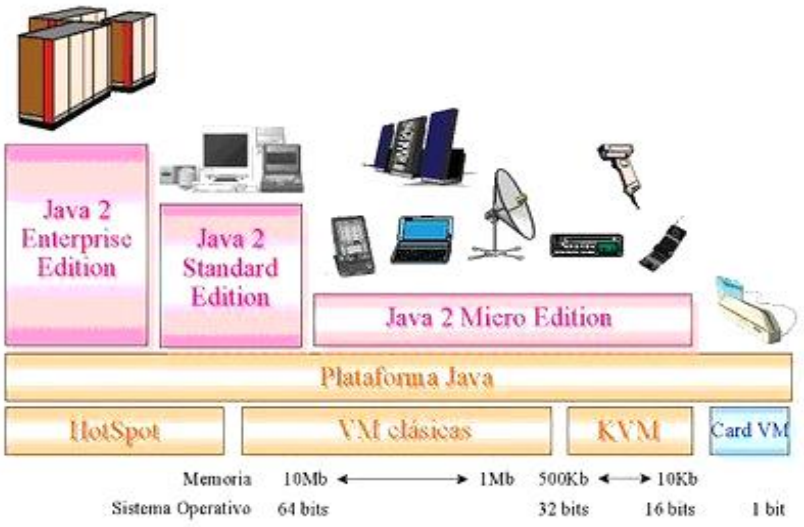


Figura 11.Esquema de J2ME. [DMB2009]

4.2.4.1 ARQUITECTURA [MUE2008]

La arquitectura de J2ME observada en la figura 12,consiste en formar una serie de APi's (Es la abreviatura de Aplication Programming Interface, un API no es más que una serie de servicios o funciones que el Sistema Operativo ofrece al programador) que permiten que cada una de las aplicaciones que se desarrollan adquieran algunas características del entorno en JAVA creando así aplicaciones para los dispositivos móviles, por eso es importante resaltar que este tipo de programación cumpla con parámetros de :

- Perfiles.
- Configuraciones.
- Máquinas Virtuales.

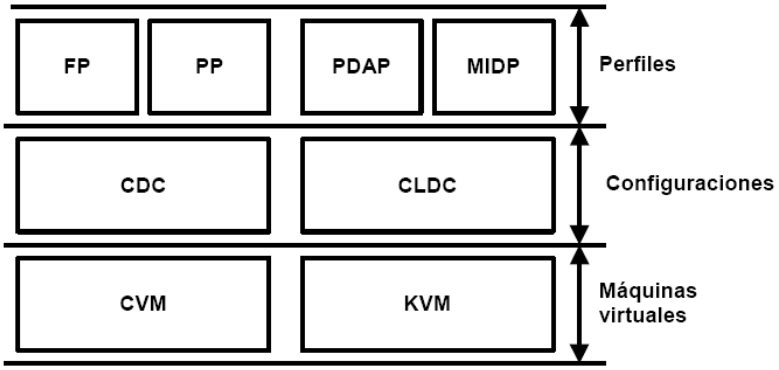


Figura 12. Arquitectura de J2ME [MUE2008]

4.2.4.2 PERFILES [PEU2008]

Los perfiles son los que se encargan de identificar a un grupo de dispositivos por la funcionalidad que tengan como son los móviles, computadores etc. y el tipo de aplicaciones que normalmente funciona en ellos.

Estos perfiles permiten que las aplicaciones realizadas sean cargadas en los diferentes dispositivos, los perfiles que existen actualmente asociados a J2ME son:

a. Fundation Profile: Este perfil define una serie de APi's sobre la CDC (Connected Device Configuration*),* utilizada sobre todo en sistemas de domótica orientadas a dispositivos que carecen de interfaz gráfica como, por ejemplo, decodificadores de televisión digital.

b. Personal Profile: Este perfil se encarga de conceder a la configuración CDC de una interfaz gráfica completa, con capacidades web y soporte de applets (una manera de incluir programas complejos en el ámbito de una página web. Estos applets se programan en Java y por tanto se benefician de la potencia de este lenguaje para la Red)

c. Mobile Information Device profile: Este perfil está orientado para dispositivos con una conectividad limitada, capacidad gráfica y una entrada de datos alfanumérica reducida.

Se ha optimizado para las pequeñas pantallas, mecanismos de introducción de datos y otras características de los dispositivos móviles, se observa en la figura 13

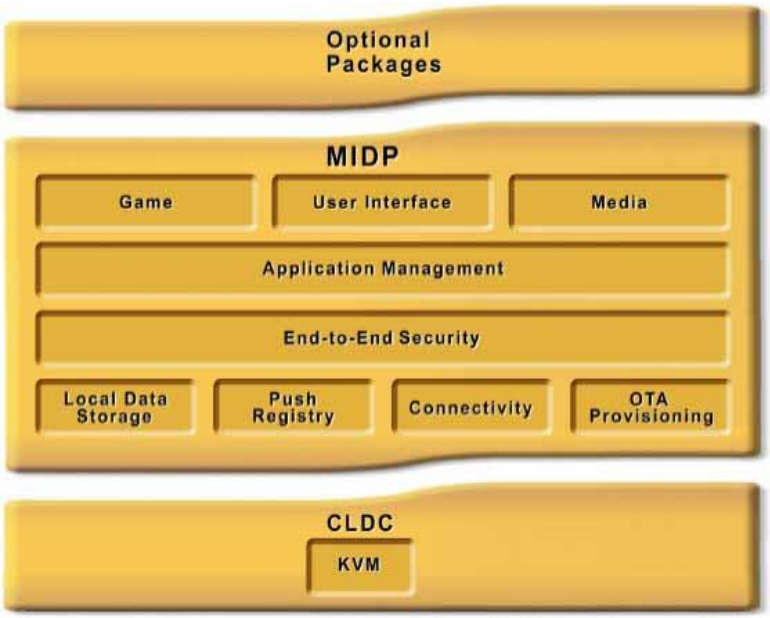


Figura 13. Perfiles MIDP [MUE2008]

4.2.4.2 CONFIGURACIONES [PEU2008]

Las configuraciones son un conjunto de APIS básicas de Java, compuestas por una máquina virtual y librerías que comparten un entorno similar, actualmente existen dos configuraciones en J2ME y son:

a. CDC (Configuración de dispositivos con conexión)

Se desarrollo para dispositivos con una memoria disponible reducida, utiliza una máquina virtual en Java similar en sus características a una de J2SE **(**Plataforma Java 2, Edición estándar**,** un kit de desarrollo de software que se utiliza para escribir applets y aplicaciones con el lenguaje de programación Java), teniendo en cuenta que tiene limitaciones en la parte gráfica y en la memoria del dispositivo.

b. CLDC (Connected Limited Device Configuration).

Configuración de dispositivos limitados con conexión, CLDC (Connected Limited Device Configuration). Este tipo de configuración fue diseñado para dispositivos con procesadores con ciertas limitaciones en memoria y en rendimiento, como:

- Teléfonos móviles celulares
- PDAs (Personal Digital Assistant o Ayudante personal digital)

Los perfiles que existen para esta configuración son:

- MIDP (Mobile Information Device Profile), usada en teléfonos móviles celulares que permite las aplicaciones gráficas
- Foundation Profile.
- Personal Profile

4.2.4.4 MÁQUINA VIRTUAL

La Máquina Virtual Java (JVM) es el entorno en el que se ejecutan los programas Java donde su función principal es la de garantizar la portabilidad de las aplicaciones en Java.

Existe un conjunto de máquinas virtuales para cada tipo de dispositivo que son:

a. KVM: es la máquina virtual más pequeña que ocupa entre 40 y 80 kilobytes de memoria, orientada así a dispositivos de baja capacidad y de memoria.

b. CVM: Compact virtual machine, orientado a procesadores de 32 bits de gama alta, y una memoria de 2MB o más memoria RAM.

4.2.5 NÚMEROS Y SECUENCIAS PSEUDOALEATORIAS

Son números o secuencias, generados por medio de una función (determinista, no aleatoria) y que aparentan ser aleatorios. Estos números pseudoaleatorios se generan a partir de un valor inicial aplicando iterativamente la función.

La sucesión de números pseudoaleatorios es sometida a diversos test para medir hasta qué punto se asemeja a una sucesión aleatoria. Y fundamentalmente las sucesiones pseudoaleatorias son más rápidas de generar que las aleatorias.

Los dos principales inconvenientes de las sucesiones de números pseudoaleatorios son que a partir de un mismo valor inicial se genera la misma sucesión y que, en general, la sucesión es periódica. Estos inconvenientes se solucionan escogiendo generadores con periodos largos.

Las secuencias pseudoaleatorias son utilizadas en diversos entornos relacionados con el mundo de las telecomunicaciones. Entre otros ámbitos de aplicación, podemos encontrar la criptografía y transmisión de datos.

4.2.6 ALGORITMO INTERNACIONAL DE CIFRADO DE DATOS IDEA [AIU2008]

IDEA (International Data Encryption Algorithm), es un algoritmo de cifrado simétrico que opera con bloques de 64 bits y una clave de 128 bits, observado en la figura 14.

Por su estructura en array (medio de guardar un conjunto de objetos de la misma clase), el proceso puede realizarse mediante sucesivas iteraciones de las operaciones realizadas en una etapa. La clave de 128 bits es dividida sucesivamente en subclaves de 16 bits repetidamente hasta obtener 52 subclaves

“Hace un uso total de 48 subclaves para ejecutar las 8 iteraciones y cuatro subclaves adicionales para la transformación final. El proceso de cifrado y descifrado utiliza el mismo esquema. A partir de la misma clave, se elige por el modo de generación de las subclaves”

.
IDEA utiliza tres operaciones matemáticas en su proceso que se realizan con grupos de 16 bits y son:

- Operación O-exclusiva (XOR) bit a bit
- Suma módulo 2^{16}
- Multiplicación módulo $2^{16}+1$

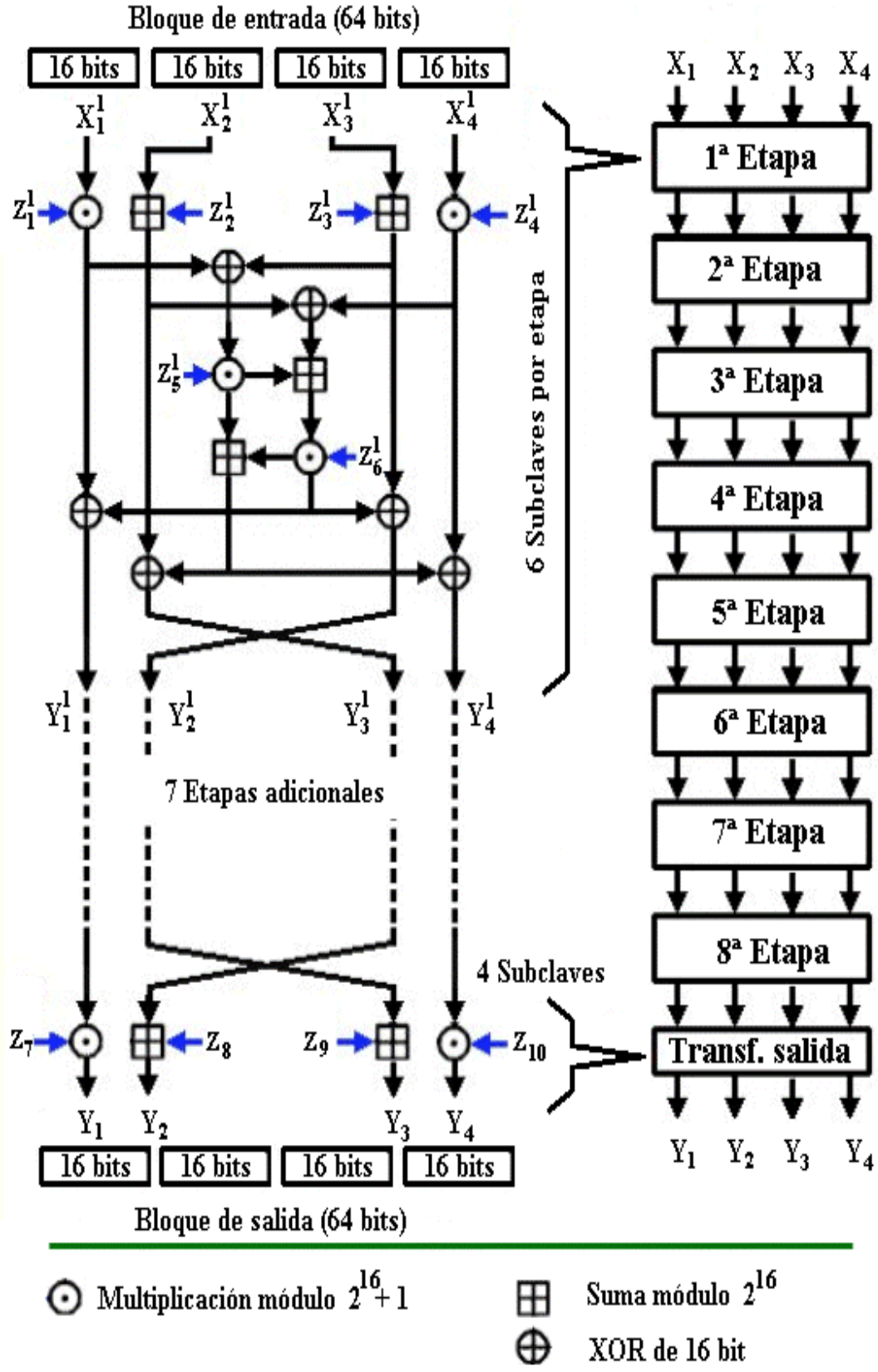


Figura 14. Algoritmo IDEA [AIU2008]

Al realizar el proceso de expansión de subclaves se elige la clave, y el resultado puede guardarse para su utilización posterior. Por esto en las implementaciones Hardware-Software del algoritmo este proceso es adecuado para ser ejecutado por el software por que requiere una pequeña fracción del tiempo de cálculo.

La estructura de cifrado permite que se realicen óptimos cambios, donde solo cambiando un bit en el texto se diferencia toda la parte de cifrado, lo que facilita la divergencia en sucesivas reiteraciones.

Cada una de las rondas usa seis subclaves de 16 bits las cuales se derivan de la clave de 128 bits (Ki $1 \leq i \leq 6$) para transformar una entrada X en cuatro bloques de 16 bits que serán las entradas de la siguiente ronda. Las salidas de la ronda 8 serán las entradas de la última ronda de transformación, que utiliza cuatro subclaves adicionales para producir el cifrado final Y = (Y1 Y2 Y3 Y4).

Para esto utiliza tres operaciones que se realizan a lo largo de las 8 rondas y de la última transformación; basadas en aritmética modular tales como:

- Multiplicación modulo (2^16)+1.
- Suma modulo 2^16.
- XOR de 16 bit.

La multiplicación modular utilizada en este algoritmo tiene unas condiciones tales como:

- Si alguna de las dos entradas a la multiplicación es 0 se deben cambiar por 2^16.
- Si el resultado es igual a 2^16 se debe cambiar por 0.

La implementación de cada una de las rondas del cifrador IDEA se realiza de acuerdo a las siguientes secuencias.

$$X_1 \leftarrow X_1 \odot K_1^{(r)}, X_4 \leftarrow X_4 \odot K_4^{(r)}, X_2 \leftarrow X_2 \boxplus K_2^{(r)}, X_3 \leftarrow X_3 \boxplus K_3^{(r)}.$$
$$t_0 \leftarrow K_5^{(r)} \odot (X_1 \oplus X_3), t_1 \leftarrow K_6^{(r)} \odot (t_0 \boxplus (X_2 \oplus X_4)), t_2 \leftarrow t_0 \boxplus t_1.$$
$$X_1 \leftarrow X_1 \oplus t_1, X_4 \leftarrow X_4 \oplus t_2, a \leftarrow X_2 \oplus t_2, X_2 \leftarrow X_3 \oplus t_1, X_3 \leftarrow a.$$

Y la implementación de la ronda de transformación según la siguiente secuencia.

$$Y_1 \leftarrow X_1 \odot K_1^{(9)}, Y_4 \leftarrow X_4 \odot K_4^{(9)}, Y_2 \leftarrow X_3 \boxplus K_2^{(9)}, Y_3 \leftarrow X_2 \boxplus K_3^{(9)}.$$

Las claves de cifrado son calculadas por medio de un proceso de corrimientos de 25 veces de la clave original de 128 bits hacia la izquierda y se rompe en 8 grupos de 16 bits; este proceso se repite hasta obtener 56 subclaves, de las cuales se utilizaran las primeras 48 para las 8 rondas (cada ronda utiliza de a 6 subclaves) y las siguientes 4 en la ronda de transformación de esta forma se dejan de utilizar las ultimas 4 subclaves.

round r	$K'^{(r)}_1$	$K'^{(r)}_2$	$K'^{(r)}_3$	$K'^{(r)}_4$	$K'^{(r)}_5$	$K'^{(r)}_6$
$r = 1$	$(K_1^{(10-r)})^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$	$(K_4^{(10-r)})^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
$2 \leq r \leq 8$	$(K_1^{(10-r)})^{-1}$	$-K_3^{(10-r)}$	$-K_2^{(10-r)}$	$(K_4^{(10-r)})^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
$r = 9$	$(K_1^{(10-r)})^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$	$(K_4^{(10-r)})^{-1}$	—	—

Tabla 3. Esquema de claves para descifrar.

Para realizar el proceso de descifrado se utiliza el mismo algoritmo pero se deben calcular las claves inversas de las utilizadas en el proceso de cifrado y organizarlas de acuerdo a la tabla 3.

Las claves inversas se hayan según un proceso donde se calcula el inverso aditivo de las que se utilizaron en el proceso de cifrado para realizar la operación suma modular. Las siguientes se hayan calculando el inverso multiplicativo, aplicando el Algoritmo Euclidiano Extendido a las claves que se utilizaron en la multiplicación modular en el proceso de cifrado y las últimas se derivan de un proceso de transposición.

Al realizar el proceso de expansión de subclaves se elige la clave, y el resultado puede guardarse para su utilización posterior. Por esto en las implementaciones Hardware-Software del algoritmo este proceso es adecuado para ser ejecutado por el software por que requiere una pequeña fracción del tiempo de cálculo.

4.2.7 TECNOLOGIAS UTILIZADAS

4.2.7.1 TELEFONO CELULAR

Figura15.Motorola A1200

En la selección del dispositivo móvil que se observa en la figura 15 influyo el trabajo realizado en el proyecto de grado CONTROL DÓMÓTICO MEDIANTE UN DISPOSITIVO MÓVIL CELULAR CON TECNOLOGIA BLUETOOTH, que en conjunto con el PARANI (modulo Bluetooth) realizaron una muy buena comunicación.

Este celular maneja una configuración que está diseñada para dispositivos con conexiones de red intermitentes, procesadores lentos y memoria limitada:

Características:

- Procesador:16 bit/16 MHz o más
- Memoria: 160-512 KB de memoria total disponible para la plataforma Java
- Alimentación: Alimentación limitada, a menudo basada en batería
- Trabajo en red: Conectividad a algún tipo de red, con ancho de banda limitado habitualmente
- Conexión Bluetooth

Maneja el MIDP (perfil para dispositivos de información móvil) que proporciona un perfil que es una versión de J2ME (Java 2 Micro Edition) integrada en el hardware de celulares relativamente modernos que permite el uso de programas Java, tales como juegos, aplicaciones o todo tipo de software. Se apoya en CLDC (Connected Limited Device Configuration) que proporciona los aspectos básicos para ejecutar Java en pequeños dispositivos, proporciona los paquetes y clases necesarios para el desarrollo de nuestras aplicaciones intuitivas y gráficas.

4.2.7.2 MODULO PARANI ESD-100

El modulo PARANI que se observa en la figura 16 se emplea para realizar comunicaciones seriales de corto alcance, además emplea la técnica de salto de frecuencia FHSS que permite reducir la interferencia. Existen dos modos de configuración, el primero mediante comandos AT ingresados por Hyperterminal y el segundo con el software propio del dispositivo. Éste modulo tiene un alcance de 100 Mts corresponde a una clasificación (Clase 1) y presenta una serie de especificaciones que están dadas en la tabla 4.

Figura 16 PARANI ESD100

Éste modulo se escogió por el buen desempeño que presentó en el proyecto de grado CONTROL DÓMOTICO MEDIANTE UN DISPOSITIVO MÓVIL CELULAR CON TECNOLOGÌA BLUETOOTH al momento de establecer comunicaciones con dispositivos Bluetooth que soporten el perfil serial RFCOMM. [PDG2007]

Características	Descripción
Interfaz serial (RS232)	Velocidades seriales desde 1200bps hasta 230400bps
Interfaz Bluetooth	Bluetooth versión 1.2 Protocolos: RFCOM, L2CAP, SDP. Perfil: Puerto Serial. Clase 1 Potencia: 18dBm Alcance: hasta 100mtrs.
Configuración	ParaniWin, Hyperterminal.
Alimentación	Alimentación: 3.3VDC Corriente: 300mA min.
Propiedades Físicas	Dimensiones: Largo: 27.5mm Ancho: 30mm Alto: 14mm Peso: 5g

Tabla 4. Especificaciones PARANI

4.2.7.2 MICROCONTROLADOR MC68HC908AP16

El microcontrolador MC68HC908AP16, es un miembro de bajo costo y con una arquitectura de alto rendimiento de la familia MC68HC08; utiliza registros para datos de 8 bits y de 16 bits para sus diferentes tipos de direccionamiento. Además posee nuevos sistemas embebidos tales como la operación multiplicación, la división y otros que le dan grandes prestaciones sobre otros microcontroladores como el PIC 16F877A. Estas prestaciones pueden ser representadas en una disminución sustancial en el tamaño del código.

Posee un juego de 118 instrucciones (28 más que el 68HC05); tiene un ciclo de bus que es un cuarto de la frecuencia del cristal; esto significa que se utilizan cuatro fases de reloj interno para la ejecución de una instrucción simple. También posee un módulo llamado PLL que brinda mejores prestaciones en este dispositivo, generando internamente una frecuencia mayor a la del cristal; además tiene una interfaz de comunicación serial que le permite comunicarse con otros dispositivos en modo full-duplex de forma asíncrona y otros módulos de los que se encontrará información más detallada en su respectivo manual.

Para su programación se utilizan las herramientas (software) CodeWarrior V 5.1 y el programador (hardware) para microcontroladores freescale V 4.0 que se observa en la figura 17 Y 18.

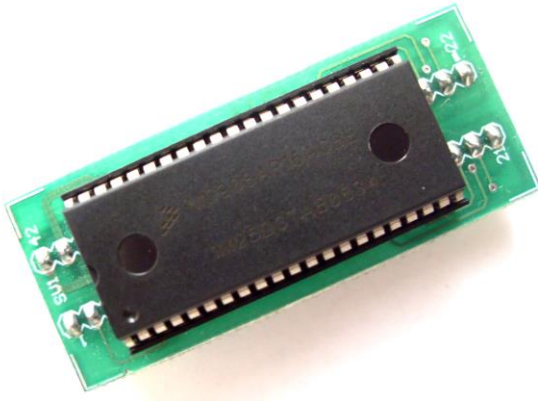


Figura 17. MC68HC908AP16

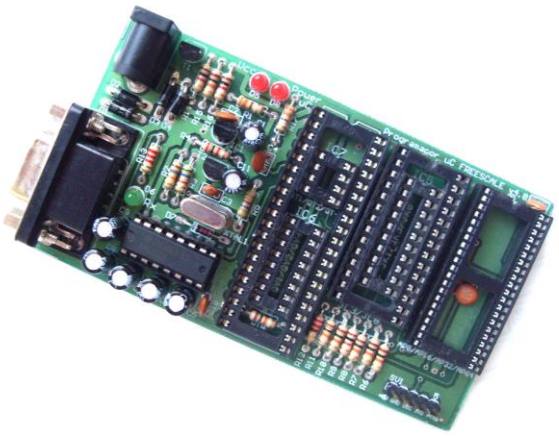


Figura 18 Programador freescale V 4.0

En esta familia se encuentran diferentes tipos de microcontroladores que están básicamente diferenciados por el tamaño de su memoria tal y como se muestra en la tabla 5.

Dispositivo	Tamaño Ram	Tamaño Memoria Flash
MC68HC908AP64	2,048	62,368
MC68HC908AP32	2,048	32,768
MC68HC908AP16	1,024	16,384
MC68HC908AP8	1,024	8,192

Tabla 5. Microcontroladores Familia MC68HC08

4.3 MARCO CONCEPTUAL

Para completar en su totalidad el desarrollo de este proyecto es necesario tener cuidado con cada una de las especificaciones técnicas, que poseen cada uno de los dispositivos y conceptos que se explicaron en el marco teórico.

Una de las partes fundamentales de este proyecto la compone el microcontrolador, ya que sin este no se puede controlar el modulo Bluetooth (Parani),por esto es necesario aprender a programar en este caso en assemmbler (tipo de lenguaje de bajo nivel utilizado para escribir programas informáticos que constituye la representación más directa del código máquina específico para cada arquitectura de computadoras legible por un programador), para su respectiva programación y la segunda que se debe tener en cuenta, son los conocimientos básicos y avanzados que se deben obtener sobre J2ME, ya que sin esta no sería posible el desarrollo de aplicaciones en el dispositivo móvil.

Un mínimo conocimiento sobre el sistema de encendido del automovil y el funcionamiento del bloqueo central, es necesario para poder controlar el flujo de corriente, que activa o desactiva la bobina que proprociona la corriente necesaria para encender cada una de las bujias del automovil y por consiguiente encender el motor.

La bobina de encendido es un elemento que actúa directamente en el proceso de encendido del automóvil, ya que convierte la tensión normal de 12 voltios proveniente de la batería, en otra de alta tensión del orden de 10 Kv a 15 Kv; que permite a las bujías producir las chispas de encendido del motor. Por lo anterior se dice que la bujía es un transformador con una relación de espiras entre el primario y secundario de 100/1 aprox., que acumula la energía eléctrica, para después transmitirla a las bujías en forma de impulso a través del distribuidor, siguiendo un orden determinado. [MEC2009]

En la actualidad hay sistemas de encendido doble que se componen de más de una bobina y que permiten activar dos o más bujías al mismo tiempo y que dependen del número de cilindros del motor debido a que por más cantidad de cilindros menor el ángulo del rotor dentro del distribuidor.

En la figura 19 se observa los componentes que integran el sistema de encendido del automóvil y de qué forma se activan durante el proceso de encendido.

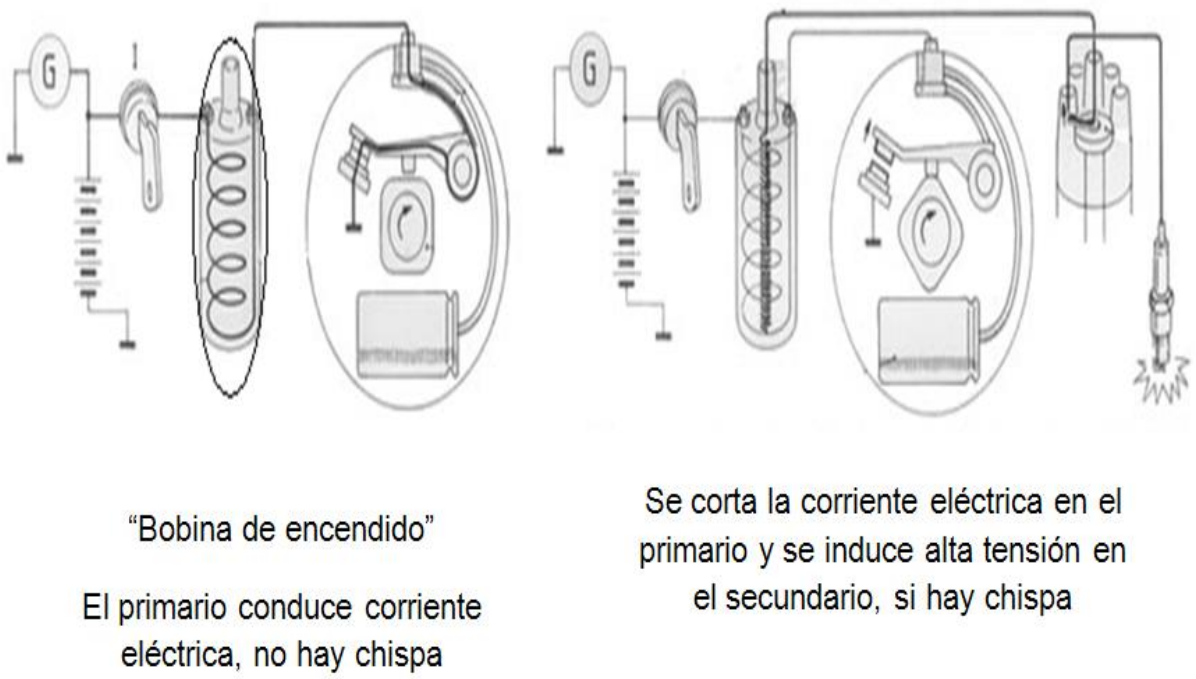


Figura 19. Sistema de encendido del automóvil. [MEC2009]

Durante el proceso de encendido del automóvil hay diferentes variaciones de voltaje y corriente en la bobina que deben tomarse en cuenta.

4.4 ESTADO DEL ARTE

4.4.1 CRIPTOGRAFÍA

La importancia que sigue teniendo la Criptografía radica, en que es el único método actual capaz de hacer cumplir el objetivo de la Seguridad Informática: "mantener la Privacidad, Integridad, Autenticidad", cumpliendo con el No Rechazo, relacionado a no poder negar la autoría y recepción de un mensaje enviado, uno de los proyectos más destacados:

Implementación de la primera red protegida con criptografía cuántica **[TEN2009]**

"Concluyen en éxito más de 4 años de investigación en el proyecto SECOQC Realizado en España, La primera red comercial de telecomunicación protegida con tecnologías de criptografía cuántica ha sido presentada en una demostración por responsables del proyecto SECOQC.

Fundamentado en los principios de la física cuántica, el sistema desarrollado hace impenetrable el intercambio de datos a cualquier tipo de escucha o intromisión no autorizada.

El proyecto SECOQC (Desarrollo de una Red Global para la Comunicación Segura Basada en la Criptografía Cuántica) de la UE, en el que han trabajado 41 socios de 12 países europeos durante cuatro años y medio, ha concluido en un rotundo éxito: el pasado 8 de octubre del 2008 se llevó a cabo una demostración en la que por primera vez una red comercial de telecomunicación ha transportado datos protegidos mediante encriptación cuántica, según se informa en un comunicado.

La seguridad en la transmisión de datos a través de una red protegida mediante criptografía cuántica está garantizada por las leyes de la física, concretamente por el principio de incertidumbre de Heissemberg, que define la imposibilidad de observar un sistema sin provocar perturbaciones en el mismo"

4.4.2 UML

- Una de las características más relevantes de la notación UML es su capacidad para absorber nueva semántica sin romper su lógica interna. La necesidad de implementar servicios web a través de complejas arquitecturas con múltiples capas de componentes y una gran dispersión geográfica se ha convertido en una gran necesidad, por esto es que se creó una extensión de la notación UML denominada **WAE "Web Application Extensión"** que permite rentabilizar toda la gramática interna de UML para modelar aplicaciones con elementos específicos de la arquitectura de un entorno WEB.

  En las páginas de juegos y comunidades se ha convertido en algo vital, tanto así que hoy día brindan cursos especializados

- "El 29 de enero de 2008 **Visual Paradigm for UML 6.2 Internacional Limited**, empezó a anunciar la comercialización de Visual Paradigm para UML (VP-UML) 6.2, para el uso de herramientas de CASE para su desempeño como instrumento de UML.

  Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML." [VIPA2009]

5. METODOLOGÍA

El proyecto de grado “APLICACIÓN DE SOFTWARE EN UN DISPOSITIVO MÒVIL CELULAR PARA LA SEGURIDAD DE UN AUTOMÓVIL MEDIANTE COMUNICACIÓN BLUETOOTH CON SISTEMA DE CIFRADO” fue desarrollado siguiendo una metodología planteada en el anteproyecto de grado, donde se tomaron en cuenta cada uno de los objetivos planteados y el grado de importancia de cada uno de estos.

El esquema de la metodología está conformado por 7 etapas consecutivas que ayudaron a desarrollar el proyecto en su totalidad.

- Etapa 1: en esta primera fase se realizó una búsqueda de información general que nos permitió documentarnos y enriquecer nuestros conocimientos sobre las tecnologías y conceptos que serian implementados posteriormente en el proyecto, luego teniendo en cuenta la información recopilada, nuestras expectativas del proyecto y los aportes del asesor técnico, se planteo el problema, la justificación y los objetivos que dieron bases firmes para la implementación del proyecto.

- Etapa 2: para el cumplimiento de los objetivos fue necesario el aprender a utilizar las diferentes herramientas de programación tales como:

  -MATLAB: necesario para comprender el comportamiento matemático del cifrador idea.

  -CODE WARRIOR: Se utiliza para programar el microcontrolador en el lenguaje Assembler.

  -NETBEANS: Utilizado para desarrollar aplicaciones en dispositivos móviles en J2ME.

  -OFFICE VISIO: Utilizado para diseñar los esquemas de modelado de sistemas UML

- Etapa 3: En esta etapa se desarrollo el algoritmo de cifrado simétrico por bloques IDEA, donde inicialmente se desarrollaron las operaciones aritméticas modulares, necesarias para el funcionamiento del algoritmo y un generador de claves pseudoaleatorias que le ofrece al cifrador las claves necesarias para cifrar o descifrar los datos.

- Etapa 4: En esta etapa se estableció la comunicación Bluetooth entre el celular y el modulo Parani ESD-100. El celular maneja un API (Aplication Programming Interface, un API no es más que una serie de servicios o funciones que el sistema operativo ofrece al programador) de comunicación basado en el API JSR82 y el modulo mediante comandos AT generados por el microcontrolador. Para el intercambio de datos fue necesario realizar un proceso de sincronización para asegurar el envió del dato cifrado(64 bits) desde el celular hacia el microcontrolador

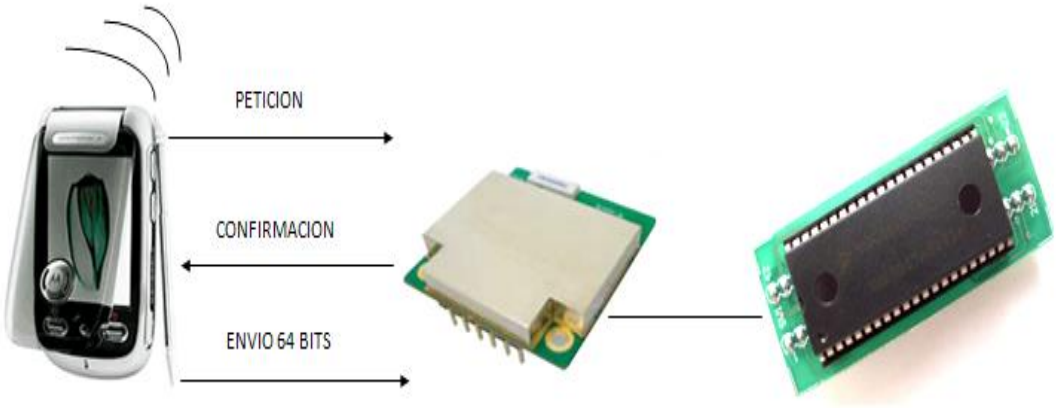


Figura 20 Esquema comunicación entre dispositivos

- Etapa 5: En esta etapa se implemento el hardware necesario para poder controlar y monitorear el sistema de seguridad del automóvil, primero se realizó el esquemático en Circuit Maker, para la simulación del circuito y finalmente se diseño en el circuito impreso.

- Etapa 6: En esta etapa se hizo el desarrollo de la monografía, donde presentamos todas las pruebas y resultados obtenidos durante el desarrollo del proyecto así como las conclusiones.

6. DESARROLLO

Para la realización del proyecto de grado APLICACIÓN DE SOFTWARE EN UN DISPOSITIVO MÓVIL CELULAR PARA LA SEGURIDAD DE UN AUTOMÓVIL MEDIANTE COMUNICACIÓN BLUETOOTH CON SISTEMA DE CIFRADO, se siguieron las siguientes etapas:

6.1 RECOPILACIÓN DE INFORMACIÓN

En esta parte se investigo y se reunió la información acerca de:

- Software de desarrollo.
- Especificación de dispositivos.

6.1.1SELECCIÓN DE SOFTWARE Y TECNOLOGÍA

Para el desarrollo se utilizaron diferentes software libres en el mercado, que nos permitieron modelar y hacer ejecución de código, para el modelado se utilizo UML (lenguaje unificado modelado), que permite la visualización y documentación del software para que las personas que no estuvieron presentes en su desarrollo puedan interpretarlo con cada uno de los diagramas que se realizan , Matlab se utilizó como una herramienta matemática permitiendo hacer una simulación de los cálculos matemáticos.

Netbeans 6.1 fue empleado para crear el entorno gráfico y funcional en el celular empleando el lenguaje de programación en Java (J2ME), para esta elección se consultó cuales son las tecnologías que permiten la programación de dispositivos móviles y que actualmente existen en el mercado. La tecnología usada en el móvil celular fue la del Motorola A1200.

Figura 21 .Software libre Netbeans 6.1

6.2 SOFTWARE

6.2.1 UML (LENGUAJE UNIFICADO MODELADO)

UML en el desarrollo del proyecto de grado se dividió en diferentes etapas:

1. Requerimientos funcionales, estos fueron:

- Ingreso a la aplicación.
- Selección de control activar bloqueo del automóvil.
- Selección de control desactivar bloqueo del automóvil.
- Selección de control activar monitoreo del automóvil.
- Selección de control desactivar monitoreo del automóvil.

2. Casos de uso y diagrama de aplicación para cada uno de los requerimientos.

3. Diagramas de clases y de secuencia.

- Es indispensable UML cuando se está elaborando un software permite construir y documentar todos los elementos que componen un sistema, ayudando comprender grandes sistemas mediante gráficos y textos.
- Permite capturar la idea de un sistema para comunicarse posteriormente a quien esté involucrado en su proceso de desarrollo, esto se lleva a cabo mediante un conjunto de símbolos y diagramas que tienen fines distintos dentro del proceso de desarrollo.

6.2.2 MATLAB

MATLAB es una herramienta matemática de computadora diseñada para hacer una gran cantidad de cálculos matemáticos. Por esto necesita que la máquina donde va a ser instalada tenga un mínimo de requerimientos como:

- Ordenador IBM PC o 100 % compatible con procesador Intel 486 o Pentium.
- Coprocesador matemático 487SX (excepto para equipos 486DX o Pentium que ya lo llevan incorporado.
- 8 Mb de memória RAM.
- Unidad de discos de alta densidad (1.4 Mb) de 3 1/2".
- Microsoft Windows 3.1 o Windows 95.
- Ratón.
- Adaptador gráfico de 8 bits (para obtener 256 colores simultáneos).
- 15 Mb. De espacio físico en disco duro (17 Mb para configuración de MATLAB Y SIMULINK).

La herramienta de MATLAB 6.5 fue utilizada para tener una vista más clara sobre el comportamiento matemático del algoritmo de cifrado IDEA; antes de encaminarse en la programación en J2ME y sobre el microcontrolador donde el algoritmo será implementado posteriormente.

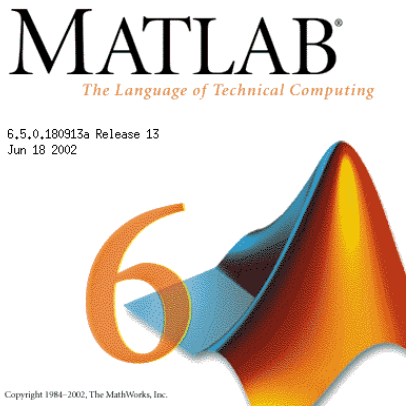


Figura 22. MATLAB 6.5

6.2.3 NETBEANS

Este software se puede descargar gratuitamente de la página de Internet http://www.netbeans.org/downloads/ , para que tenga un correcto funcionamiento, el computador donde se instale debe de tener:

- mínimo 512Kb de memoria.
- espacio de disco de 200Mb.

Cuando se realice la instalación se descarga el software jdk-6u1-windows-i586-p que permite correr programas en Java y se instala igualmente el software sun_java_wireless_toolkit-2_5_2, que permite trabajar con dispositivos móviles.

La elaboración de este software tiene 2 etapas una interfaz gráfica de usuario y el desarrollo de un API de Bluetooth que se describen a continuación:

6.2.4 INTERFAZ GRÁFICA

En el desarrollo de la interfaz gráfica se tuvo en cuenta que fuera de fácil uso para el usuario, los pasos a seguir fueron:

1. Solicitud de contraseña.
2. Pantalla de bienvenida.
3. Pantalla de conexión con el modulo Parani ESD100.
4. Pantalla de búsqueda de conexión.
5. Pantalla de conexión exitosa con logo de la Universidad.
6. Menú de selección.
7. Selección.
8. Confirmación de la selección.

6.2.4.1 Solicitud de contraseña

Esta consiste en la petición de una contraseña al usuario que debe de conocer para acceder a la aplicación de forma correcta, esta contraseña se compara con una que ya previamente fue cargada en el programa, si la contraseña es incorrecta se enviará al usuario a una pantalla de error, pero si esta es correcta avanzará a la siguiente pantalla.

6.2.4.2 Pantalla de bienvenida

Esta consiste en la bienvenida al control de seguridad del automóvil, donde se visualiza una imagen del automóvil donde se tenga instalada la aplicación, la opción de “OK” es para que continúe a la pantalla siguiente o “BACK” que permitirá devolverse una ventana.

6.2.4.3 Pantalla de conexión con el modulo Parani ESD100

Esta consiste en el inicio de la conexión con el modulo Parani ESD100 que cuenta con una Mac (dirección de control de acceso al medio) especifica, cuando el usuario pulsa “OK” se activará el Bluetooth para que empiece el proceso, o “BACK” que le permitirá devolverse una ventana.

6.2.4.4 Pantalla de búsqueda de conexión

Esta consiste en el proceso de búsqueda de conexión con el modulo Parani ESD100 residente en el automóvil.

6.2.4.5 Pantalla de conexión exitosa con logo de la Universidad

Esta consiste en visualizar la imagen de la Fundación Universitaria San Martín, donde se configura para que sea mostrada por 4 segundos.

6.2.4.6 Menú de selección

En esta pantalla se muestra el menú de selección el cual consiste en el control de bloqueo y control de monitoreo, cuando uno de estos dos se selecciona se da “OK” para seguir a la selección dada, también posee la opción de “EXIT” para salir del programa.

6.2.4.7 Selección

Dependiendo de la selección inicial que haya escogido el usuario se realiza una comunicación con el modulo receptor por medio del API que controla el Bluetooth.

6.2.4.8 Confirmación de la Selección

El usuario visualizara la confirmación cuando se realice la acción seleccionada, el tiempo de espera máximo es de 5 segundos.

6.3 COMUNICACIÓN CON EL API BLUETOOTH

En el desarrollo realizado se tuvo en cuenta inicialmente la comunicación por Bluetooth, donde la programación del dispositivo móvil utiliza un API del estándar JSR-82, que depende de la configuración CLDC de J2ME utilizada en el paquete javax.bluetooth, que permite proveer la funcionalidad para la realización de búsquedas de dispositivos, búsquedas de servicios y comunicación.

Un cliente Bluetooth deberá realizar las siguientes operaciones para comunicarse con un servidor Bluetooth:

• Búsqueda de dispositivos.

• Búsqueda de servicios.

• Establecimiento de la conexión.

• Comunicación.

Se creó una clase llamada LocalDevice que representa el dispositivo en el que se está ejecutando la aplicación, esta se obtiene mediante LocalDevice.getLocalDevice (). Este objeto permite obtener información sobre el dispositivo (modulo Parani):

- modo de conectividad.
- dirección Bluetooth y nombre del dispositivo.

El objeto DiscoveryAgent es el que permite realizar cancelar búsquedas de dispositivos y de servicios. Y también nos servirá para obtener listas de dispositivos ya conocidos.

La clase RemoteDevice representa un dispositivo remoto y tiene métodos similares a LocalDevice que, representa al dispositivo en el que se ejecuta la aplicación. De esta forma se obtiene su dirección Bluetooth mediante getBluetoothAddress ().

Se crea una clase BTUtility, la cual se utiliza para implementar las etapas de descubrimiento de dispositivos, obtener un número de servicio UUID en la que se implementa el descubrimiento de dispositivos.

La UUID (número identificador universal cuya valor representa algún servicio) en este caso se utiliza el 0X1101, este número permite utilizar el SPP.

Una vez inicializado el Bluetooth del dispositivo se busca el puerto serie (0x1101), el UUID y el atributo por medio del comando discoveryAgent.searchServices.
Una vez hecha esta se procede a pasar a la aplicación, donde se busca un SPP, este será utilizado para habilitar el envió de datos, en esta parte solo se busca y se almacena para su posterior utilización.

Ahora se procede a buscar dispositivos de Bluetooth cercanos, esta se hace por la configuración del comando startInquiry, lo que se hace con esta es que se comunique únicamente con la dirección del modulo Parani ESD-100, cuando realiza la búsqueda y es exitosa genera un InquiríCompleted y si no es exitosa visualiza un error y vuelve a comenzar la búsqueda.

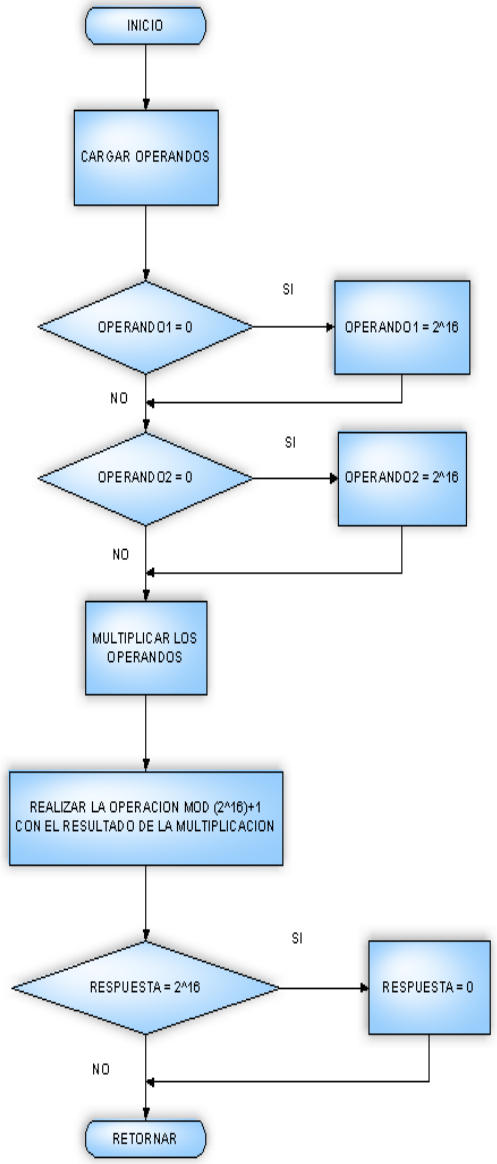
Cuando se descubre algún servicio se nos notificará a través del objeto DiscoveryListener mediante el método servicesDiscovered(), que permite buscar los servicios en un dispositivos mediante un ciclo, y se tomara el ultimo que este aparezca al final de este, dado a que lo almacena varias veces y se ubica en la última posición.

En el momento que el usuario selecciona alguna de las opciones del menú que son: activar monitoreo, desactivar monitoreo, activar control monitoreo y desactivar control monitoreo, se lleva a cabo el algoritmo de criptografía IDEA, donde se ingresa un dato de 64 bits cifrado, después se toman los últimos 8 bits de cada trama y este es el dato que se envía por Bluetooth, cuando este dato lo recibe el microcontrolador lo descifra y realiza la acción respectiva en el automóvil.

Lo anterior se implementa cuando se llama a la clase cifrador, en esta clase se encuentran los métodos: cifrar, descifrar, generador de claves y las operaciones matemáticas: multiplicación, suma y xor; todo este procedimiento es un algoritmo de cifrado aritmético por bloques con una clave de cifrado de 128 bits y 52 subclaves de cuatro bloques que están compuestas por 16 bits, generando finalmente 8 rondas computacionalmente idénticas.

6.4 CONFIGURACIÓN DEL MICROCONTROLADOR

6.4.1.1 MULTIPLICACIÓN



Lo primero que se desarrollo en la programación del microcontrolador fue la elaboración de las operaciones que se realizan en las 8 rondas y en la última transformación; basadas en aritmética modular como:

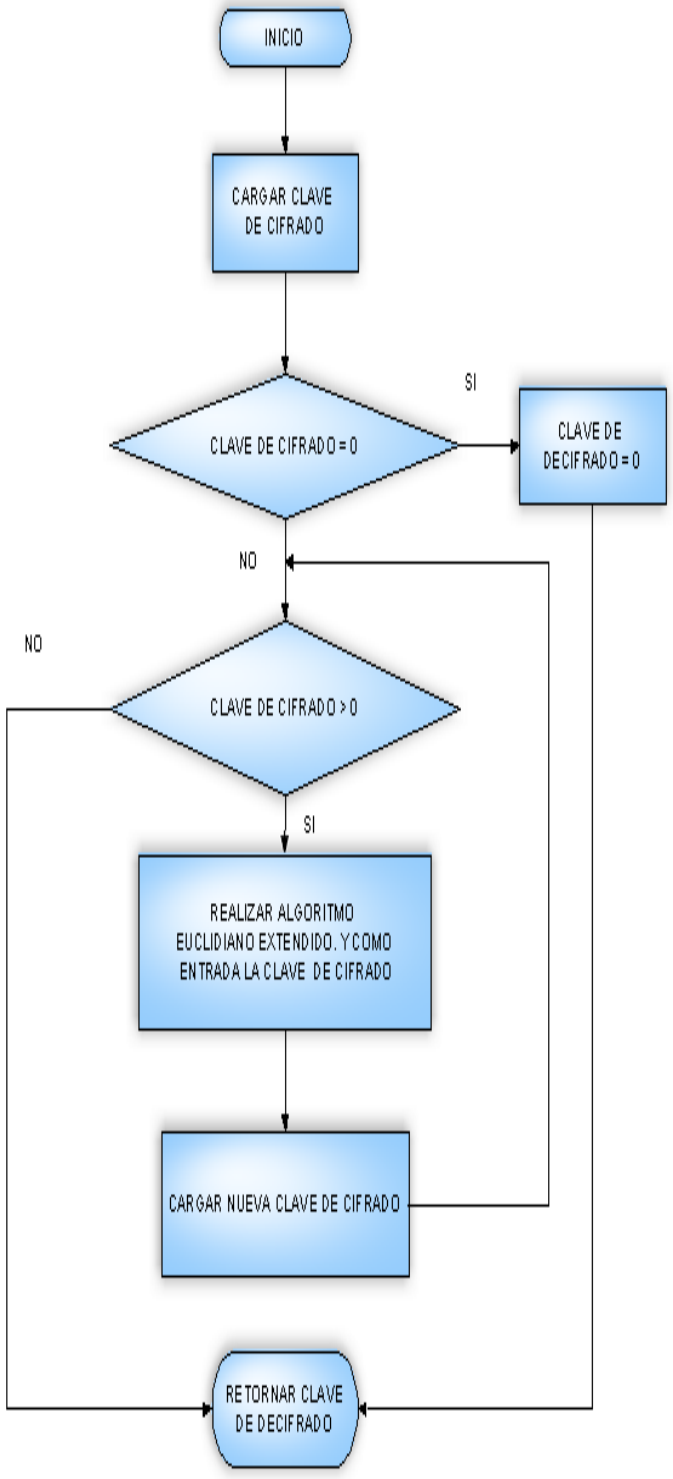
- Multiplicación
- Xor
- Suma

La multiplicación por ser una operación especial, se presenta mediante un diagrama de flujo, para mostrar la secuencia de la programación que se realizó para su implementación, como se muestra en el diagrama de la izquierda.

La suma es una operación con modulo 2^16.

La Xor es una operación de 16 bit.

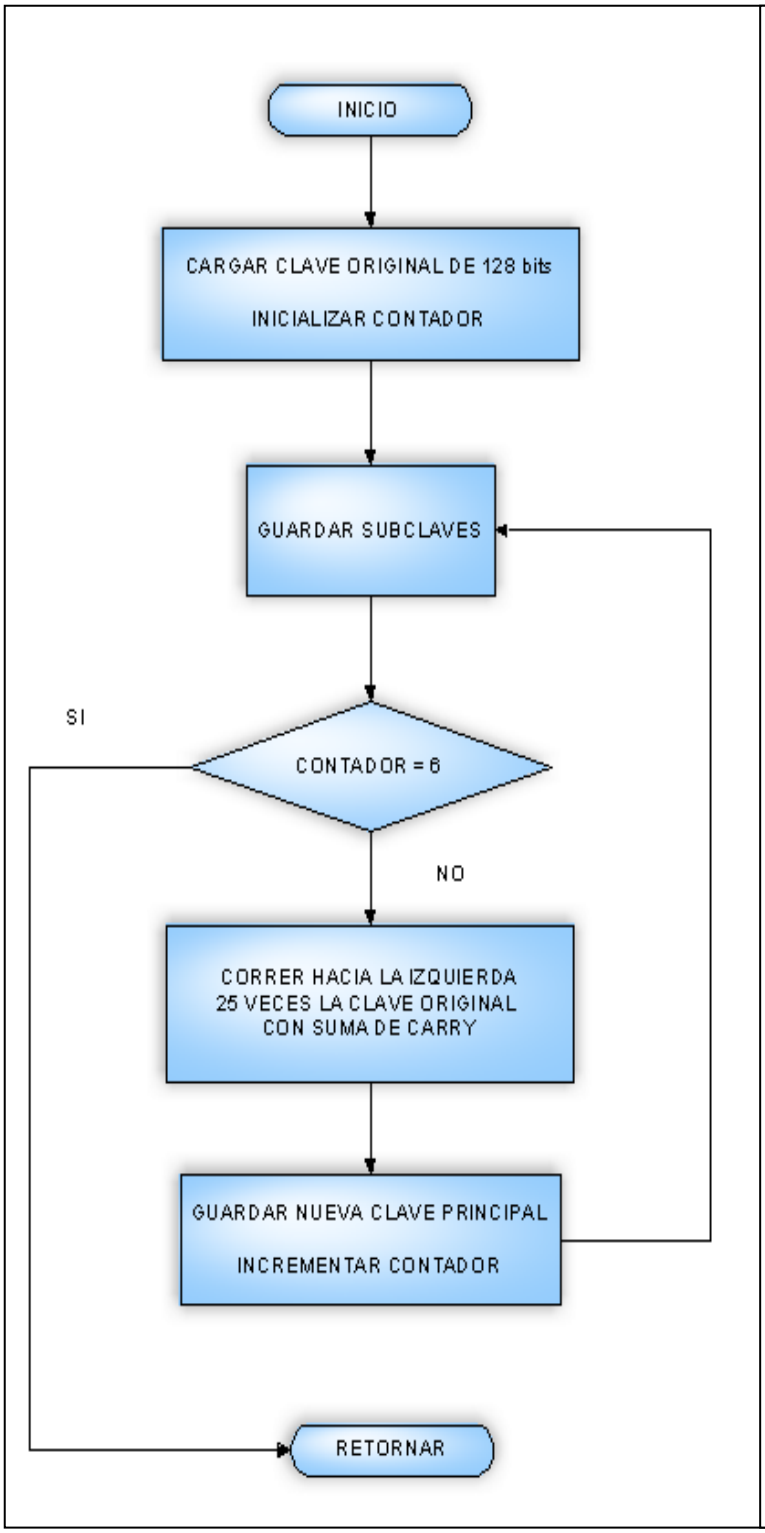
6.4.1.2 INVERSO MULTIPLICATIVO



El cifrador IDEA, para el proceso de descifrado utiliza las claves del generador para realizar dicho proceso, estas claves se hallan mediante un proceso de transposición de claves, inverso aditivo e inverso multiplicativo de una clave que se halla mediante la implementación del algoritmo euclidiano extendido:

Método para encontrar el máximo factor común de dos enteros, donde se divide el número mayor entre el número menor, se repite la división, utilizando el residuo como divisor, hasta que el residuo se convierte en cero, así el último residuo diferente a cero es el máximo factor común de los dos enteros

6.4.1.3 GENERADOR DE CLAVES DE CIFRADO



El cifrador IDEA, contiene una clave general de 128 bits de la cual generaran 52 claves para el cifrado de los datos y 52 claves para el descifrado.

El diagrama de la izquierda nos muestra la secuencia que se debe seguir para obtener las 52 claves de cifrado,

6.4.2 ESQUEMA DEL DISEÑO DEL HARDWARE PARA LA COMUNICACIÓN Y CONTROL DEL SISTEMA DE SEGURIDAD DEL AUTOMÓVIL

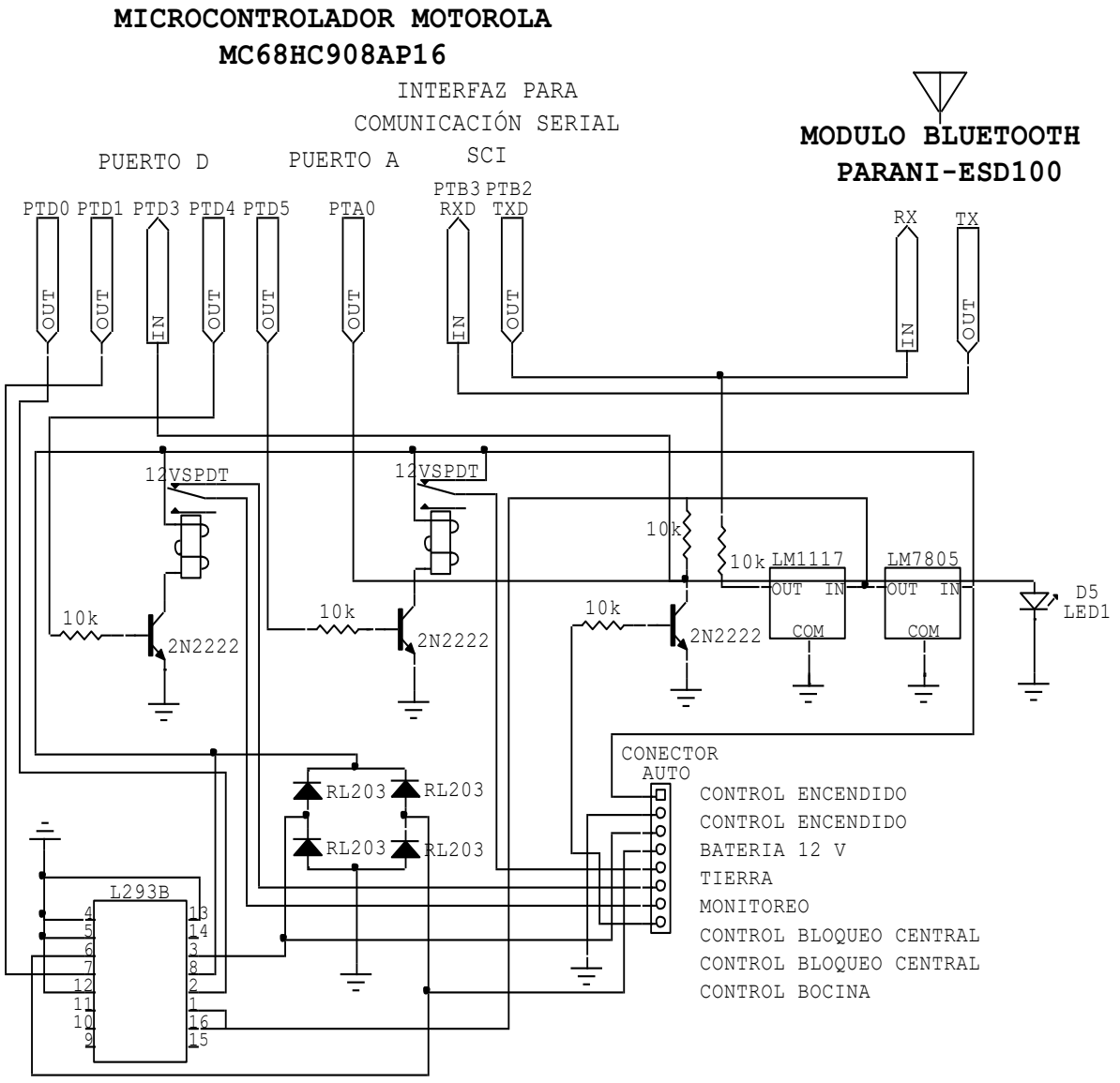


FIGURA 23.Diseño de comunicación y control de sistema de seguridad del automóvil.

6.4.3 DIAGRAMA EN BLOQUES DE LA COMPOSICIÓN DEL PROYECTO DE GRADO

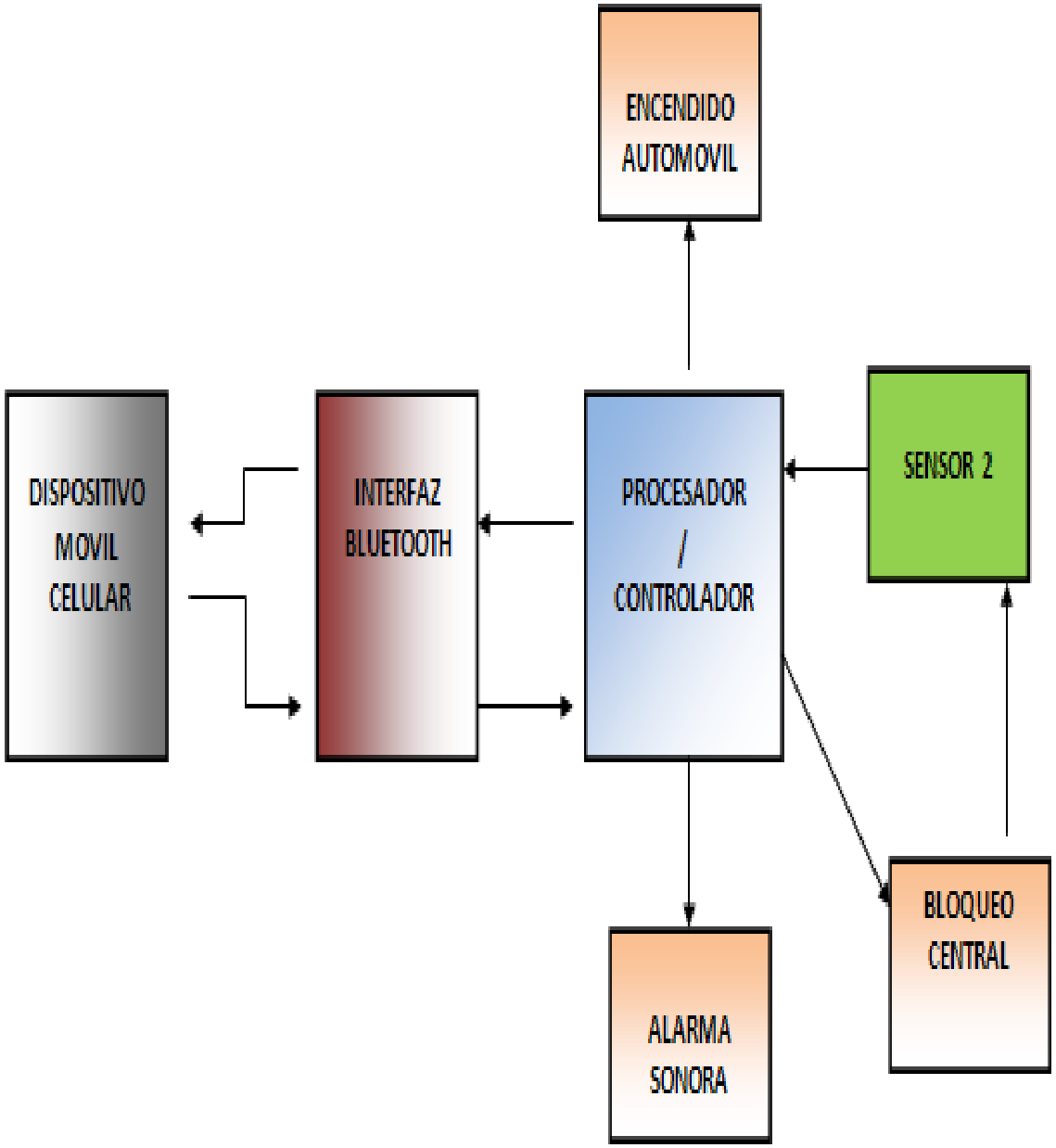


FIGURA 24. Diagrama en bloques

6.4.4 CONEXIÓN DEL PROYECTO DE GRADO EN EL AUTOMÓVIL

En la figura 25 se muestra un diseño de cómo se realizaron las conexiones para el correcto funcionamiento del circuito, que se desarrolló en el proyecto de grado. La alimentación de 12 voltios del circuito se tomó directamente de la batería, luego por medio del circuito se realizó un puente en el cable de baja tensión de la bobina, para que por medio de un relé se hiciera control del encendido del automóvil. El control sobre la bocina de la alarma, se realizó simplemente controlando el paso de corriente eléctrica hacia su modulo de control al igual que el encendido, con el uso de un relé. Por último se conectaron al circuito, tres líneas del bloqueo central. La primera en color morado de la figura 25 para poder monitorear si las puertas están abiertas o no, y las de color verde para poder controlar el bloqueo central desde el circuito del proyecto.

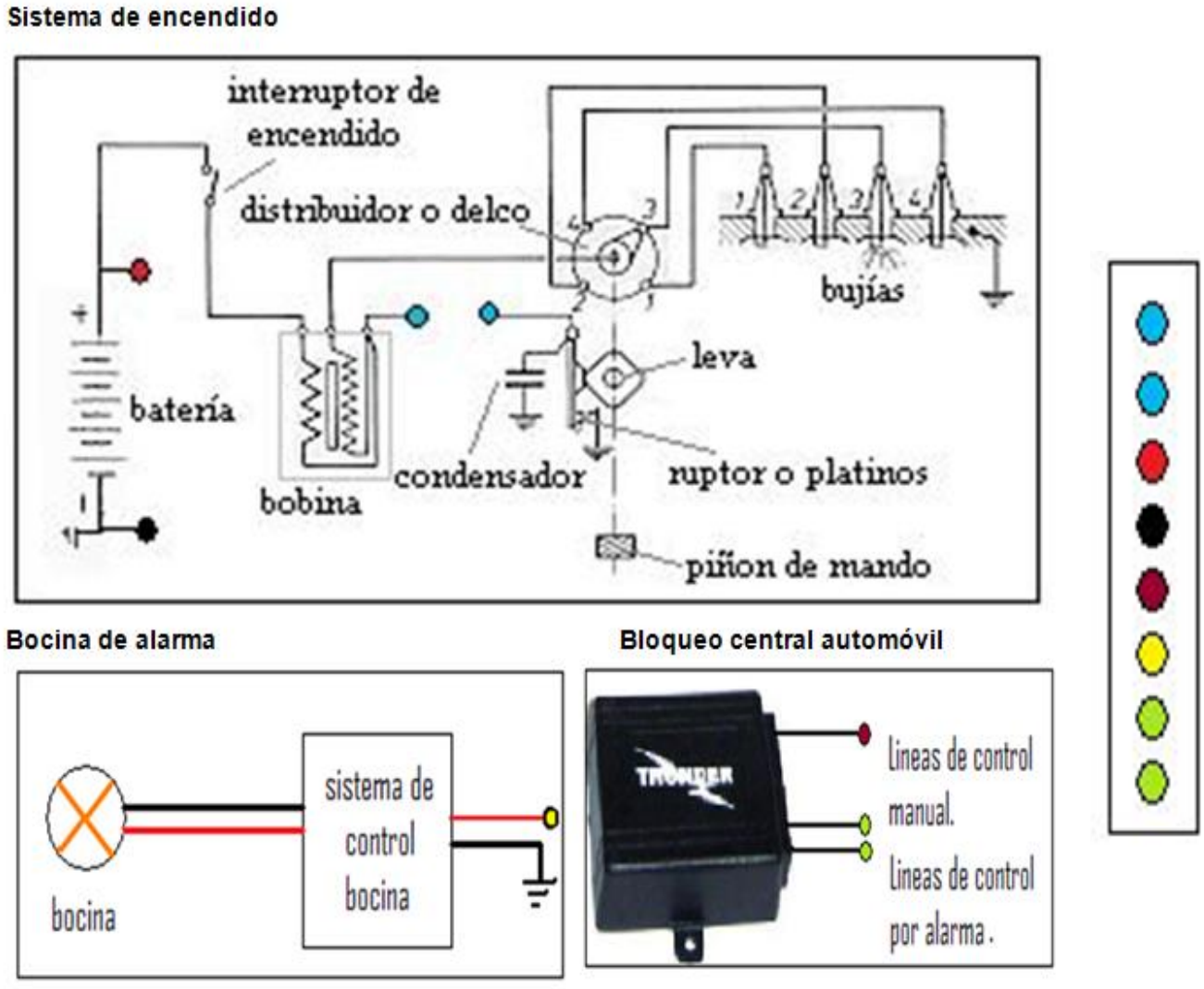


Figura 25. Conexiones Proyecto de grado – Automóvil.

7. PRUEBAS Y RESULTADOS

7.1 PRUEBA DE DISTANCIA

Los dispositivos Bluetooth utilizados para el proceso de la conexión fueron:

*Celular Motorola A1200: Clase 1(10m)
*Modulo Parani ESD-100: Clase 2(100m)

DISTANCIA Metros	1 m	2 m	3 m	4 m	5 m	6 m	7 m	8 m	9 m	10 m	11 m	12 m	13 m	14 m
Estado de la conexión con línea de vista	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	NO
Estado de la conexión con el circuito ya instalado en el automóvil	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	NO	NO	NO	NO

Tabla 6. Mediciones

7.2 PRUEBA CON HYPERTERMINAL

El uso de la herramienta Hyperterminal de Microsoft fue indispensable para el desarrollo de la conexión, la transmisión y recepción de datos por esto se desarrollaron las siguientes pruebas:

*Prueba de cifrado y descifrado de datos en el microcontrolador

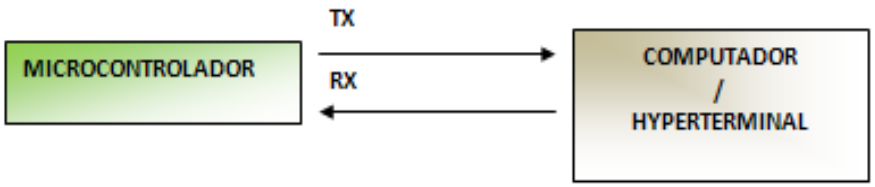


Figura 26.Prueba con Hyperterminal

Primero se desarrollo el algoritmo de cifrado IDEA en el microcontrolador, luego se enviaron los datos que se querían cifrar del Hyperterminal al microcontrolador, este lo cifraba y los devolvía al Hyperterminal, visualizando en la pantalla del computador, luego los descifraba y los volvía a enviar al Hyperterminal y los visualizaba en la pantalla del computador.

7.3 PRUEBA CON COMANDOS AT

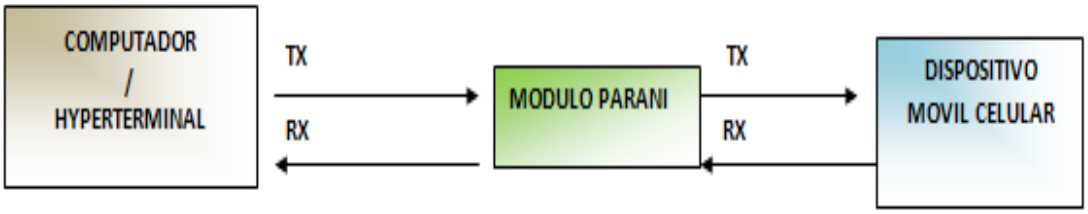


Figura 27. Transmisión y Recepción

Después de tener establecido el API de comunicación BLUETOOTH en el celular, se realizo conexión con Hyperterminal donde se enviaron los comandos AT al modulo para establecer la comunicación BLUETOOTH con el celular, enviando así los datos que se querían cifrar al celular; estos datos se reenviaban al modulo y los visualizaba en la pantalla del Hyperterminal, después de este procedimiento el celular los descifra y los volvía a enviar al Hyperterminal para poder ser visualizados finalmente.

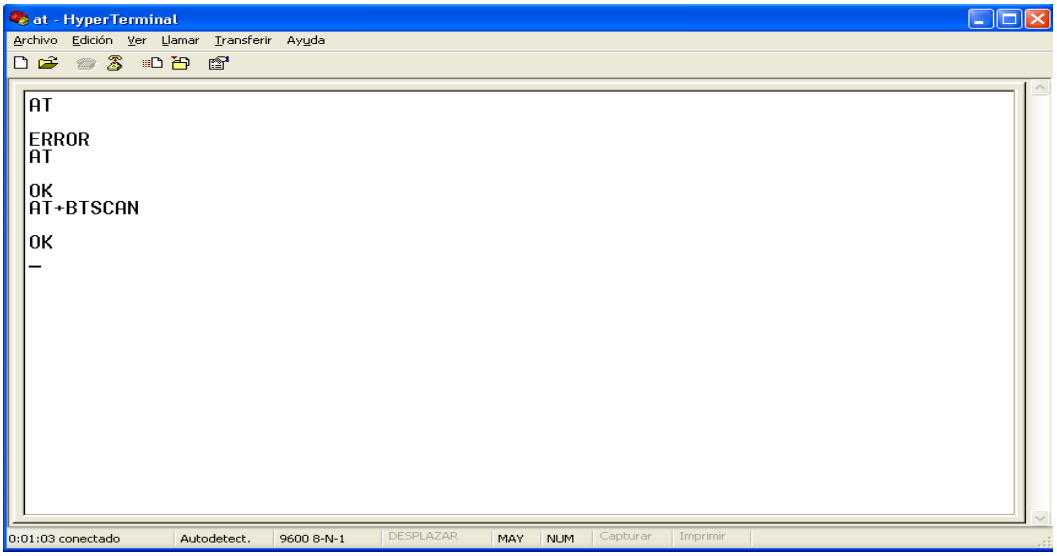


Figura 28. Comandos AT

7.4 PRUEBA DE HARDWARE

Se instalo un switch en el cable de la bobina del automóvil para comprobar si se podía aislar la corriente que controla la bobina del encendido luego de este procedimiento se simularon las señales de la alarma, que van al bloqueo central para poder abrir y cerrar las puertas.

Ya instalado el circuito en el automóvil se realizó la conexión con el celular a 9m de distancia.

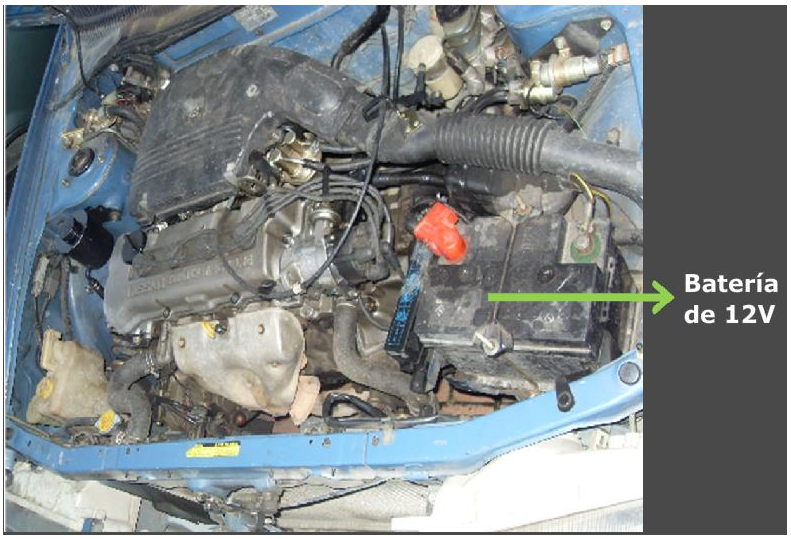


Figura 29. Fotografía Batería del automóvil

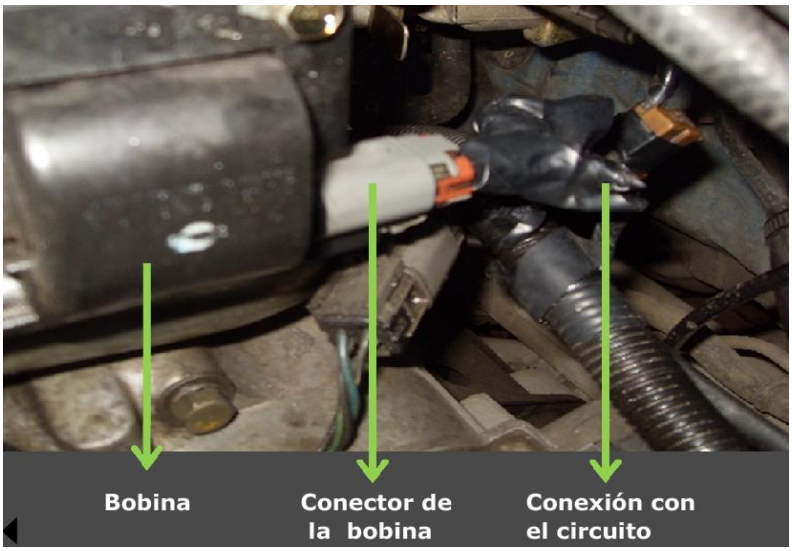


Figura 30. Fotografía Bobina del automóvil

7.5 SIMULACIONES

Estas simulaciones fueron realizadas en Wirelles toolkit, este permite ver la elaboración del entorno gráfico de desarrollo.

La figura 31 muestra el primer pantallazo dónde se inicia la aplicación que se desarrollo, este caso tiene el nombre de BLUETOOTH3.

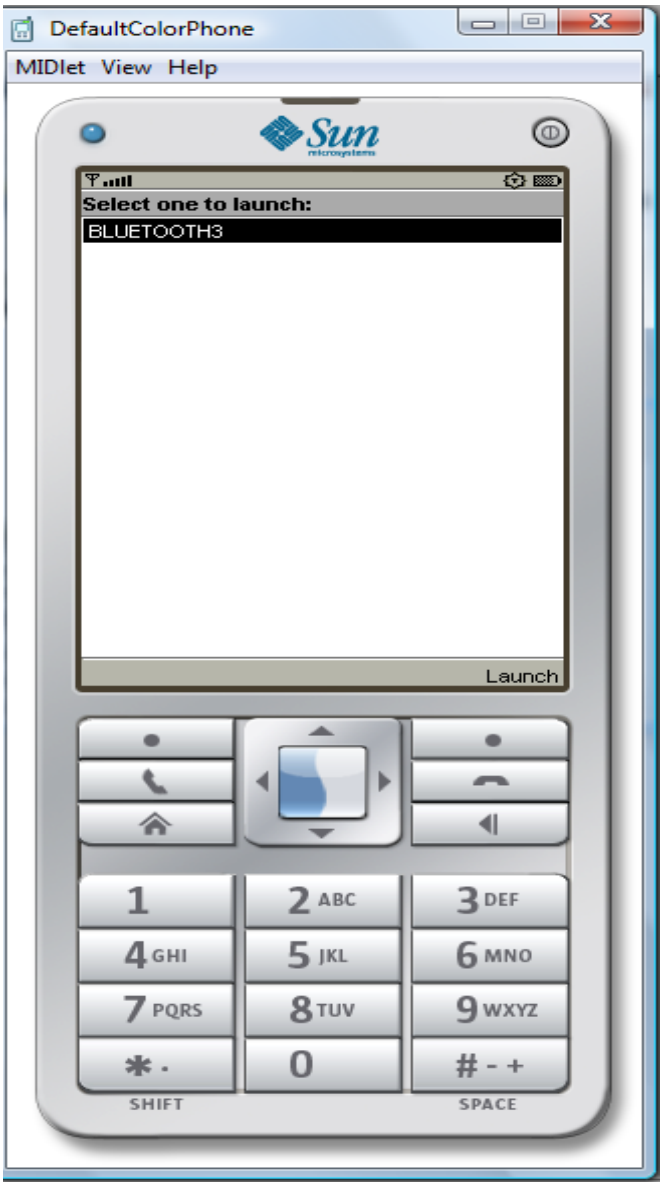


Figura 31. Inicio de aplicación

La figura 32 muestra el formulario de autenticación cuando ingresa el usuario a la aplicación

Figura 32. Formulario de autenticación

En la figura 33 se muestra la bienvenida a la aplicación.

Figura 33. Bienvenida a la aplicación

En la figura 34 se muestra inicio de conexión con el modulo Bluetooth Parani.

Figura 34.Conexión con modulo Parani

En la figura 35 se muestra búsqueda del modulo Bluetooth Parani

Figura 35. Búsqueda modulo Bluetooth Parani

Figura 36. En esta figura se visualiza el logo de la Fundación Universitaria San Martin indicando que la conexión fue exitosa.

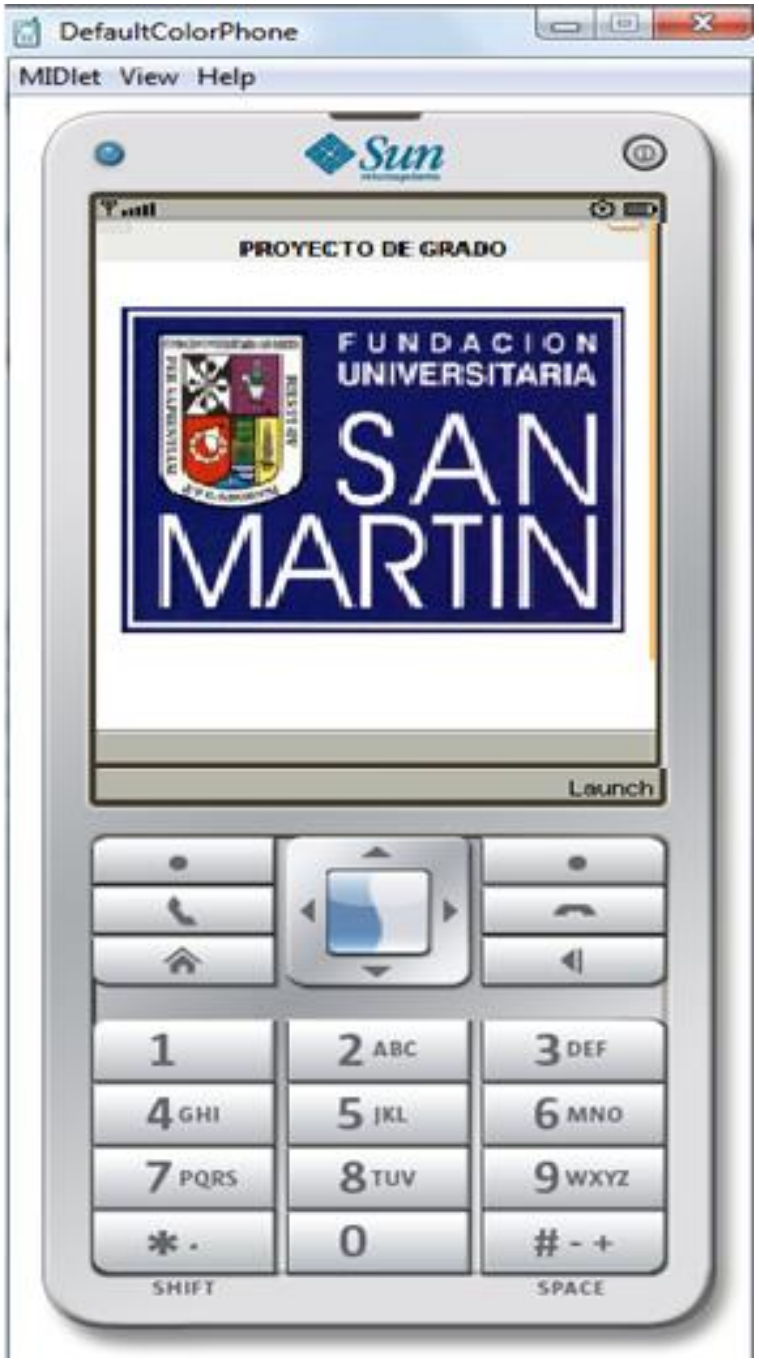


Figura 36.Éxito de Conexión

Figura 37. En esta figura se observa el menú inicial de la aplicación

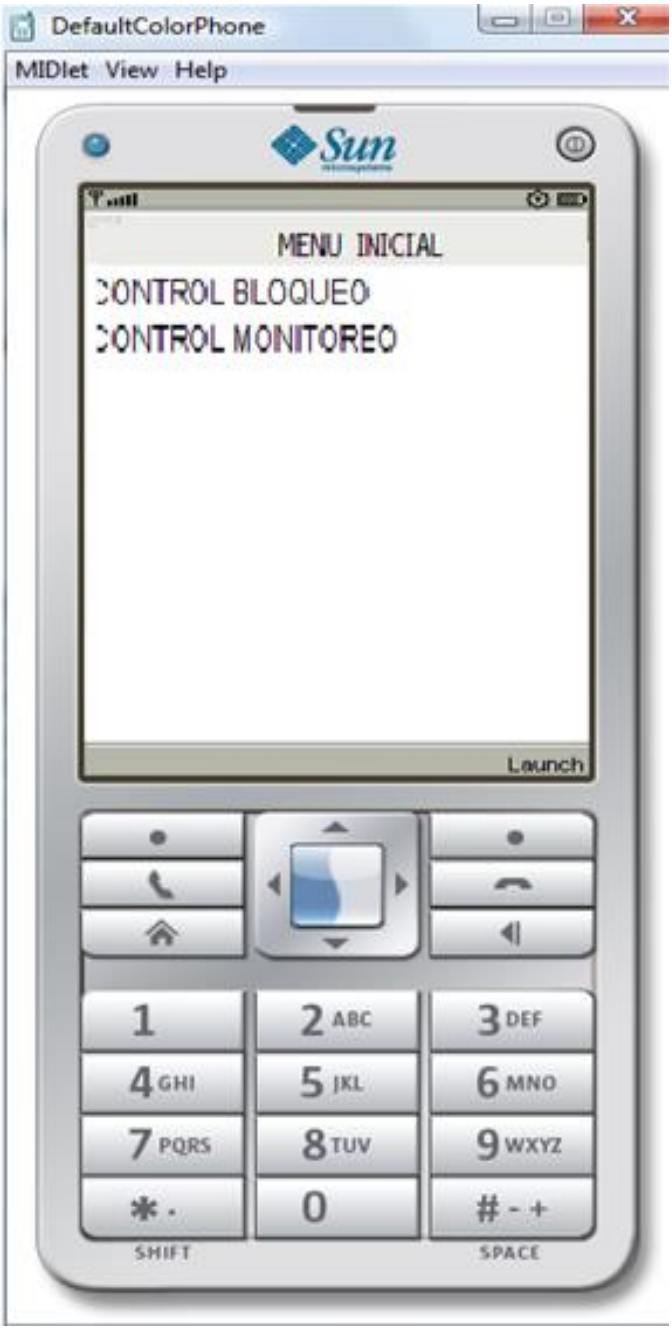


Figura 37. Menú de la aplicación

Figura 38. Se visualiza el menú de control de bloqueo

Figura 38. Menú control de bloqueo

Figura 39. Se visualiza el menú de control de monitoreo

Figura 39. Menú de monitoreo

Figura 40. Confirmación de la acción realizada

Figura 40.Confirmación

8. CONCLUSIONES

- El campo de desarrollo para dispositivos móviles es bastante amplio y hay diversas utilidades por desarrollar.

- El manejo de microcontroladores para el desarrollo de algoritmos de cifrado es muy limitado por eso se hace necesario desarrollarlo en hardware que se programe en un lenguaje de alto nivel.

- Se deberían abrir diferentes espacios académicos en la carrera de ingeniería electrónica relacionada con proyectos de dispositivos móviles y criptografía.

- Se debe tener en cuenta al momento de elegir el protocolo de comunicación, la cantidad o el tipo de datos que se quiere transmitir.

- El uso de UML es de vital importancia en el desarrollo de un software ya que permite al ingeniero tener una visión clara y concreta de cada uno de los procedimientos que se llevaron a cabo, haciendo más fácil un proceso de actualización o corrección de errores

- El manejo de la criptografía permite hacer desarrollos de aplicaciones que necesiten ofrecer seguridad y fiabilidad en la transmisión de datos, por esto se convierte en una herramienta de vital importancia para los ingenieros de hoy y del futuro.

9. RECOMENDACIONES

1. Al instalar el software en un dispositivo móvil celular se debe de tener en cuenta:

    - Procesador: 16 bit/16 MHz o más.
    - Memoria: 250 kb de memoria total disponible para la plataforma Java.
    - Alimentación: Alimentación limitada, a menudo basada en batería.
    - Trabaje la Interfaz de Programación de Aplicaciones JSR82 que permite correr aplicaciones en J2ME con conexión Bluetooth.

2. El automóvil donde se implemente el hardware debe de tener un sistema de encendido por bobina.

3. Se debe de tener en cuenta que el carro debe de tener una correcta alimentación de energía.

10. TRABAJO FUTURO

Este proyecto de grado deja una puerta abierta para desarrollos en la industria automovilística así como en la automatización industrial. Hay numerosos proyectos que se pueden derivar de este, como por ejemplo, darle menor tiempo de conexión y envió de datos entre un dispositivo de control, desarrollado bajo tecnología zeg-bee y el automóvil.

Otra propuesta que se deja a estudio, es la de implementar un sistema de envió de gran cantidad de datos entre el celular y el modulo Bluetooth.

11. GLOSARIO

Bluetooth: tecnología de radio de corto alcance que permita la conectividad entre equipos remotos. [DCI2008]

J2ME: Java 2 micro edition. Plataforma java para dispositivos móviles. [MUE2008]

UML: Conjunto de herramientas, que permite modelar (analizar y diseñar) sistemas orientados a objetos. [DISE2000]

API: Es la abreviatura de Aplication Programming Interface. Un API no es más que una serie de servicios o funciones que el Sistema Operativo ofrece al programador. [RCC2008]

HYPERTERMINAL: aplicación que puede utilizar para conectar su ordenador a otros sistemas remotos. Estos sistemas incluyen otros equipos, los sistemas de tablón de anuncios, servidores, sitios Telnet, y los servicios en línea. [TCH2009]

PROTOCOLO: Los protocolos son reglas de comunicación que permiten el flujo de información entre computadoras distintas que manejan lenguajes distintos. [RCC2008]

CIFRAR: es la ciencia de cifrar y descifrar información mediante técnicas especiales y es empleada frecuentemente para permitir un intercambio de mensajes que sólo puedan ser leídos por personas o sistemas a los que van dirigidos y que poseen los medios para descifrarlos.

RS232: El puerto serie RS-232C, presente en todos los ordenadores actuales, es la forma más comúnmente usada para realizar transmisiones de datos entre ordenadores, es una norma o estándar mundial que rige los parámetros de uno de los modos de comunicación serial. Por medio de este protocolo se estandarizan las velocidades de transferencia de datos, la forma de control que utiliza dicha transferencia, los niveles de voltajes utilizados, el tipo de cable permitido, las distancias entre equipos, los conectores, etc. [EUSK2009]

12. BIBLIOGRAFÍA

12.1Referencias Bibliográficas

- [PDG2007] Proyecto de grado realizado el segundo semestre del 2007 por Omar Ancizar Niño Ramos y Camilo Armando Sánchez Guzmán, Consultado primer semestre del 2008

- [DISE2000]Diseño orientado a Objetos (Raúl Alarcón 2000- Ed. Grupo Eidos) Consultada primer semestre del 2008

- [JAPI2002]Java APIs for Bluetooth Wireless Technology (JSR 82) Specificarion Version 1.0a - Motorola. Wireless Software, Applications & Services April 5, 2002

12.2Referencias de Internet

[FRVM2008] Universidad Tecnológica Nacional Facultad Regional Villa María. Tecnología Electrónica. http://www.frvm.utn.edu.ar/carrera/grado/electronica/tecnologia/documentos/telemando.pdf Navegada primer semestre del 2008.

[AIU2008]http://arantxa.ii.uam.es/~igonzale/recursos/IDEA_JBits_jcra2002.pdf Navegada primer semestre del 2008

[DCI2008]http://www.datafull.com/infotech/gif/medianas/bluetooth/05chica.gif Comparación con IR Navegada primer semestre del 2008

[SBT2008]http://spanish.bluetooth.com/Bluetooth/Technology/Works/Overview_of_Operation.htm Navegada primer semestre del 2008

[ULX2008]http://usuarios.lycos.es/XESC2000/Projectes/2494/tecno.html#index Navegada primer semestre del 2008

[DIM2008]http://www.datafull.com/infotech/gif/medianas/bluetooth/05chica.gif Comparación con IR Navegada primer semestre del 2008

[CCC2008]http://computacion.cs.cinvestav.mx/~jjangel/todos/cifrado_simetrico.pdf Navegada primer semestre del 2008

[EYE2008]http://www.estrellateyarde.es/discover/encriptacion Navegada primer semestre del 2008

[PEU2008]http://profesores.elo.utfsm.cl/~agv/elo323/2s06/projects/LoyolaCastillo/tecnologia.htm. Navegada primer semestre del 2008

[MUE2008]http://www.mobigame.uah.es/2003/cd/ii-ctm/ij2meesquema.pdf

[AIU2008]http://arantxa.ii.uam.es/~igonzale/recursos/IDEA_JBits_jcra2002.pdf Navegada primer semestre del 2008

[RCC2008]http://www.rastersoft.com/OS2/CURSO/APIEXPL.HTMNavegada primer semestre del 2008

[DCI2008]http://www.datafull.com/infotech/gif/medianas/bluetooth/05chica.gif Comparación con IR Navegada primer semestre del 2008

[DMB2009] http://desarrollomoviles.blogspot.com/2008/11/y-llegamos-al-j2me.html Navegada primer semestre del 2009

[CRP2009] http://www.criptored.upm.es/criptored.htm Navegada primer semestre del 2009

[EUSK2009]http://www.euskalnet.net/shizuka/rs232.htm Navegada primer semestre del 2009

[TCH2009]http://es.techfaq.com/hyperterminal.shtml&prev=hp&rurl=translate.google.com Navegada primer semestre del 2009

[BEN2009]http://bluehack.elhacker.net/proyectos/vulnerabilidades/bluebug/bluebug.html Navegada primer semestre del 2009

[TEN2009]http://www.tendencias21.net/Se-implementa-la-primera-red-protegida-con-criptografia-cuantica_a2624.html Navegada primer semestre del 2009

[MEC2009]http://mecanicavirtual.iespana.es/encend_convencional.htm Navegada primer semestre del 2009

[YOP2009]http://www.yoreparo.com/foros/electronica_automotriz/soluciones/alarma-contra-luz-encendida-t237002.htmlNavegada primer semestre del 2009

[VIPA2009]http://www.visual-paradigm.com/product/vpuml/communityedition.jsp Navegada primer semestre del 2009

ANEXO No 1 ESPECIFICACIÓN DE CASOS DE USO Y SU REALIZACIÓN

ANEXO No 2 DIAGRAMACIÓN EN UML

ANEXO No 3 CODIGO IMPLEMENTADO EN EL MICROCONTROLADOR

ANEXO No 4 CODIGO IMPLEMENTADO EN JAVA (J2ME)

ANEXO No 5 ESPECIFICACIÓN JSR82