DISPOSITIVO ELECTRONICO PARA LA REALIZACIÓN DE TRANSACCIONES BANCARIAS POR MEDIO DE UNA RED INALÁMBRICA *iDEN*

DANIEL FELIPE PINILLA GARCIA
MANUEL FELIPE RODRÍGUEZ ARRIETA
NELSON JULIÁN VILLARREAL HIGUERA

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA ELECTRÓNICA
BOGOTÁ D.C.
2005

DISPOSITIVO ELECTRONICO PARA LA REALIZACIÓN DE TRANSACCIONES BANCARIAS POR MEDIO DE UNA RED INALÁMBRICA *iDEN*

DANIEL FELIPE PINILLA GARCIA
MANUEL FELIPE RODRÍGUEZ ARRIETA
NELSON JULIÁN VILLARREAL HIGUERA

Informe Final del Trabajo de Grado
Para optar al título de Ingenieros Electrónicos

DIRECTOR

Ing. RICARDO TRIANA RUBIANO

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA ELECTRÓNICA
BOGOTÁ D.C.
2005

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA ELECTRÓNICA

RECTOR MAGNIFICO: R.P. GERARDO REMOLINA S.J.

DECANO ACADÉMICO: Ing. FRANCISCO JAVIER REBOLLEDO MUÑOZ

DECANO DEL MEDIO UNIVERSITARIO: R.P. ANTONIO JOSÉ SARMIENTO NOVA S.J.

DIRECTOR DE CARRERA: Ing. JUAN CARLOS GIRALDO CARVAJAL

DIRECTOR DEL PROYECTO: Ing. RICARDO TRIANA RUBIANO

ARTICULO 23 DE LA RESOLUCIÓN No. 13 DE JUNIO DE 1946

"La universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado.

Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque los trabajos no contengan ataques o polémicas puramente personales. Antes bien, que se vea en ellos el anhelo de buscar la verdad y la justicia".

Agradecimientos

Proyectos como este, que tienen un alto grado de complejidad, requieren del apoyo y la colaboración de mucha gente.

Queremos agradecer primero que todo a nuestras familias por ser el apoyo principal en todos los momentos, incluyendo los difíciles.

También queremos dar un agradecimiento muy especial a las siguientes personas, de las cuales obtuvimos una colaboración indispensable para el desarrollo del proyecto: Ing. Ricardo Triana Director del proyecto, Ing. Carlos Córdoba, Ing. Teodosio Varela, Ing. Alberto Acosta, Ing. John Freddy Sánchez, Erika Franco, Luz Helena Cardona, Ing. Leonardo Ramírez y todas aquellas personas que de una u otra persona influyeron en Wi-P.O.S..

TABLA DE CONTENIDO

1. INTRODUCCIÓN.....16
2. OBJETIVOS.....18
3. ESPECIFICACIONES DEL DISPOSITIVO.....19
   - 3.1. DIAGRAMAS DE FLUJO DE LOS PROCESOS NECESARIOS PARA LLEVAR A CABO UNA TRANSACCIÓN DE PAGO CON TARJETA CRÉDITO O DÉBITO USANDO WI-P.O.S. 20
     - 3.1.1. Diagrama de proceso para guardar llaves de cifrado y terminal ...20
     - 3.1.2. Diagrama de flujo de carga de parámetros iniciales.....24
     - 3.1.3. Diagrama de flujo de la realización de una transacción de pago con tarjeta de crédito o débito.....25
   - 3.2. ESPECIFICACIONES DE WI-P.O.S.....29
     - 3.2.1. Entradas y salidas del dispositivo.....29
     - 3.2.2. Componentes del dispositivo.....31
     - 3.2.3. Módulos del dispositivo.....32
   - 3.3. PROTOCOLO DE MENSAJES PARA REALIZACIÓN DE LA TRANSACCIÓN.....33
     - 3.3.1. Solicitud de parámetros iniciales.....34
     - 3.3.2. Respuesta de parámetros iniciales.....34
     - 3.3.3. Solicitud de parámetros de tarjeta.....35
     - 3.3.4. Respuesta de parámetros de tarjeta.....35
     - 3.3.5. Solicitud de transacción de pago.....37
     - 3.3.6. Respuesta de transacción de pago.....38
     - 3.3.7. Petición de transacción de reverso.....39
     - 3.3.8. Respuesta de transacción de reverso.....39
     - 3.3.9. Solicitud de llave de cifrado de PIN.....39
     - 3.3.10. Respuesta de llave de cifrado de PIN.....39

3.4. MANEJO DE LLAVES DE CIFRADO.....40
3.4.1. Manejo de Llave Maestra.....40
3.4.2. Llave de Transporte.....41
3.4.3. Llave de cifrado de PIN.....42

4. DESARROLLO.....43
4.1. MANEJO DE PERIFÉRICOS Y PROCESAMIENTO DE DATOS.....43
4.1.1. Comunicación con el teléfono móvil.....44
4.1.2. Comunicación con la impresora.....46
4.1.3. Lector de banda magnética.....46
4.1.4. Teclado para ingreso de datos.....49
4.1.5. Comunicación con el modulo de cifrado.....50
4.1.6. Rutinas del microcontrolador.....54
4.2. Cifrado de datos..... 61
4.2.1. Módulo Serie-Paralelo.....64
4.2.2. Módulo Paralelo-Serie.....67
4.2.3. Modulo DES.....68
4.2.4. Funciones de Tablas.....69
4.2.5. Funciones Específicas.....70
4.2.6. Entradas del módulo.....72
4.2.7. Integración de los módulos.....73
4.2.8. Módulo de Cifrado.....74
4.3. APLICACIÓN EN J2ME PARA EL TELÉFONO MÓVIL MOTOROLA.....75
4.3.1. Descripción de la aplicación.....76
4.3.2. Clases desarrolladas en la aplicación.....77
4.3.3. Descripción de las aplicaciones principales.....80
4.4. APLICACIÓN EN VISUAL BASIC.....89

4.4.1. Enviar Llaves.....90
4.4.2. Llave de Transporte.....91
4.4.3. Llave Maestra.....93
4.4.4. Seleccionar Llave Maestra.....95
4.4.5. Dígitos de Chequeo.....97
4.4.6. Número de terminal.....97
4.4.7. Vector de Inicio.....98
4.4.8. Reinicio de consecutivo.....99
4.5. DISEÑO DEL CIRCUITO IMPRESO.....100
5. PRUEBAS Y ANÁLISIS DE RESULTADOS.....102
5.1. VERIFICACIÓN DE FUNCIONAMIENTO DEL MÓDULO DE CIFRADO.....102
5.1.1. Pruebas de verificación de cifrado/descifrado de un bloque sin vector de inicio.....102
5.1.2. Pruebas de verificación de funcionamiento de cifrado en modo CBC.....103
5.2. PRUEBAS DE LA APLICACIÓN REALIZADA EN VISUAL BASIC.....103
5.2.1. Prueba de la carga de llave de transporte y verificación por dígitos de chequeo.....103
5.2.2. Prueba De la carga de llaves maestras y verificación por dígitos de chequeo.....104
5.2.3. Prueba de Selección de llave maestra.....105
5.2.4. Prueba de carga de número de terminal.....105
5.2.5. Prueba de la introducción de vector de inicio.....105
5.2.6. Prueba de Reinicio de número de consecutivo.....105
5.3. PRUEBAS DE FUNCIONAMIENTO DEL DISPOSITIVO REALIZANDO TRANSACCIONES BANCARIAS.....106

5.3.1. Pruebas del proceso de carga de parámetros iniciales.....106
5.3.2. Pruebas proceso de transacción de pago 108
5.3.3. Pruebas con los posibles errores de una transacción de pago.....109
5.4. ANÁLISIS DE RESULTADOS.....110
5.5. COSTOS DEL PROYECTO.....112
6. CONCLUSIONES.....114
7. BIBLIOGRAFIA.....116
ANEXO 1.....117
ANEXO 2.....135

LISTA DE FIGURAS

FIGURA 1. ESQUEMA GENERAL DE TRANSACCIÓN CON SERVIDOR DE VISA 19
FIGURA 2. ESQUEMA DE ENTRADAS Y SALIDAS DE WI-P.O.S.....29
FIGURA 3, PROCESO DE CARGA DE LLAVES MAESTRAS.....41
FIGURA 4. CONEXIÓN DEL PUERTO SERIAL USART.....46
FIGURA 5. CONEXIÓN DEL TECLADO. 50
FIGURA 6 TRANSMISIÓN DE UN 85H SEGUIDO DE UN ABH.....51
FIGURA 7. CONEXIÓN CON EL MÓDULO DE CIFRADO.....52
FIGURA 8. CIFRADO USANDO MODO CBC (CIPHER BLOCK CHAINING) B1, B2, B3 SON BLOQUES DE.....53
FIGURA 9. DESCIFRADO USANDO MODO CBC (CPHER BLOCK CHAINING).....54
FIGURA 10. PROCESO PARA LLAMAR LAS RUTINAS DEL MICROCONTROLADOR.....55
FIGURA 11. CONFIGURACIÓN SERIAL PASIVA ENTRE LA MEMORIA EPC2 Y LA FPGA ACEX 1K.....62
FIGURA 12. CONECTOR JTAG PARA PROGRAMAR LA ACEX1K.....63

FIGURA 13. MÓDULOS DE ENTRADA Y SALUDA A DES.....64
FIGURA 14. MÓDULO SERIE-PARALELO.....65
FIGURA 15. SIMULACIÓN DE MÓDULO SERIE-PARALELO CON RECEPCIÓN DE DATOS A CIFRAR.....66
FIGURA 16. SIMULACIÓN DE MÓDULO SERIE-PARALELO CON RECEPCIÓN DE LLAVE.....66
FIGURA 17. MÓDULO PARALELO-SERIE.....68
FIGURA 18. SIMULACIÓN DEL MÓDULO PARALELO-SERIE.....68
FIGURA 19. ENTRADAS Y SALIDAS DEL MODULO DES.....73
FIGURA 20. INTERCONEXIÓN ENTRE MÓDULOS.....75
FIGURA 21. PANTALLA INICIAL.....80
FIGURA 22. CARGA DE PARÁMETROS EN PROCESO.....81
FIGURA 23. CARGA DE PARÁMETROS INICIALES CARGA DE PARÁMETROS CORRECTA.....81
FIGURA 24. CARGA DE PARÁMETROS INICIALES, CARGA DE PARÁMETROS INCORRECTA.....81
FIGURA 25. CARGA DE PARÁMETROS INICIALES, ERROR EN EL PUERTO SERIAL 82
FIGURA 26. CARGA DE PARÁMETROS INICIALES, ERROR EN LA CONEXIÓN A INTERNET.....82
FIGURA 27. CARGA DE PARÁMETROS INICIALES, ERROR SIN SOLUCIÓN.....83
FIGURA 28. TRANSACCIÓN DE PAGO, PANTALLA INICIAL.....83
FIGURA 29. TRANSACCIÓN DE PAGO, PARÁMETROS INICIALES NO HAN SIDO CARGADOS.....84
FIGURA 30. TRANSACCIÓN DE PAGO, PANTALLA DE BIENVENIDA.....84
FIGURA 31. TRANSACCIÓN DE PAGO, DESLIZAR TARJETA.....84
FIGURA 32. ERROR DE TARJETA.....85

FIGURA 33. TRANSACCIÓN DE PAGO, VERIFICANDO DATOS.....85
FIGURA 34. VERIFICACIÓN DE DATOS.....86
FIGURA 35. NÚMERO INVÁLIDO.....86
FIGURA 36. PERSISTE ERROR DE NÚMERO INVALIDO.....86
FIGURA 37. INGRESE FECHA.....87
FIGURA 38. FECHA INVÁLIDA.....87
FIGURA 39. TARJETA VENCIDA.....87
FIGURA 40. INGRESO CLAVE.....87
FIGURA 41. SELECCIÓN DE TARJETA.....88
FIGURA 42. VALOR TRANSACCIÓN.....88
FIGURA 43. VALOR DE PROPINA.....88
FIGURA 44. NÚMERO DE CUOTAS.....88
FIGURA 45. TRANSACCIÓN EN PROCESO.....89
FIGURA 46. PANTALLAS DE RESULTADO DE TRANSACCIÓN.....89
FIGURA 47. PANTALLA PRINCIPAL DE LA HERRAMIENTA DE ADMINISTRACIÓN DE WI-P.O.S.....90
FIGURA 48. PANTALLA DE ENVIAR LLAVES.....91
FIGURA 49. PANTALLA DE ENVÍO DE LLAVE DE TRANSPORTE.....91
FIGURA 50. PROTOCOLO DE ENVÍO DE LLAVE DE TRANSPORTE.....92
FIGURA 51. ERRORES PRODUCIDOS AL INGRESAR MAL LOS DATOS DE LLAVE DE TRANSPORTE.....92
FIGURA 52. ENVÍO DE LA LLAVE DE TRANSPORTE EXITOSA.....93
FIGURA 53. ERROR EN LA COMUNICACIÓN ENTRE WI-P.O.S Y EL COMPUTADOR.....93
FIGURA 54. SELECCIÓN DE NÚMERO DE LLAVES TOTALES.....94
FIGURA 55. ENVÍO DE NÚMERO TOTAL DE LLAVES MAESTRAS.....94
FIGURA 56. ENVÍO DE LLAVE MAESTRA.....95
FIGURA 57. COMUNICACIÓN ENTRE WI-P.O.S Y COMPUTADOR PARA

ENVÍO DE LLAVES MAESTRAS.....95
FIGURA 58. SELECCIÓN DE LLAVE MAESTRA.....96
FIGURA 59. CONFIRMACIÓN DE ACTUALIZACIÓN DEL PROGRAMA.....96
FIGURA 60. VENTANA DE DÍGITOS DE CHEQUEO.....97
FIGURA 61. VENTANA DE ENVÍO DE NÚMERO DE TERMINAL.....98
FIGURA 62. VENTANA DE ENVÍO DE VECTOR DE INICIO.....99
FIGURA 63. COMUNICACIÓN ENTRE WI-P.O.S Y COMPUTADOR PARA ENVÍO DE VECTOR DE INICIO.....99
FIGURA 64. VENTANA DE ENVÍO DE VECTOR DE INICIO.....100
FIGURA 65. FOTOGRAFÍA DEL CIRCUITO IMPRESO.....101
FIGURA 1A. ESQUEMA GENERAL DEL PROCESO DE CIFRADO.....119
FIGURA 2A, CALCULO DE F(R, K).....122
FIGURA 3A . CÁLCULO DE LAS 16 SUBLLAVES.....126
FIGURA 4A. TRAMA DE DATOS DE TARJETAS DE BANDA MAGNÉTICA TIPO PISTA II.....129
FIGURA 5A. TRAMA DE DATOS DE TARJETAS DE BANDA MAGNÉTICA TIPO PISTA I.....129
FIGURA 6A. LECTOR DE BANDA MAGNETICA.....130
FIGURA 7A. COMPONENTES DE UNA TRANSACCIÓN BANCARIA.....133
FIGURA 8A . ADMINISTRADOR DE MENSAJERÍA ISO8385/BASE24.....134

LISTA DE TABLAS

TABLA I. CONEXIONES ENTE ACEX1K Y JTAG EN MODO SERIAL PASIVO.....63
TABLA II. NÚMEROS DE TERMINAL CONSIGNADOS EN LA BASE DE DATOS Y SUS PARÁMETROS CORRESPONDIENTES.....107
TABLA III. COSTOS DEL PROYECTO.....112

TABLA 1A. IP (PERMUTACIÓN INICIAL).....120
TABLA 2A. PERMUTACIÓN INICIAL INVERSA.....120
TABLA 3A. TABLA DE EXPANSIÓN E.....122
TABLA 4A. TABLA DE SUSTITUCIÓN S1.....123
TABLA 5A. TABLAS DE SUSTITUCIÓN S1, S2, S3 Y S4.....124
TABLA 6A. TABLAS DE SUSTITUCIÓN S5, S6, S7 Y S8.....124
TABLA 7A. TABLA DE PERMUTACIÓN P.....125
TABLA 8A. TABLA DE PERMUTACIÓN PC-1.....126
TABLA 9A. CORRIMIENTOS A LA IZQUIERDA DE LOS BLOQUES C_N Y D_N SEGÚN EL NÚMERO DE LA ITERACIÓN.....127
TABLA 10A. TABLA DE PERMUTACIÓN PC-2.....127
TABLA 1B. INFORMACIÓN ASOCIADA A LOS NÚMEROS DE LOS TERMINALES.....140
TABLA 2B. USUARIOS INGRESADOS EN LA BASE DE DATOS Y SUS CORRESPONDIENTES PARÁMETROS.....143
TABLA 3B. PARÁMETROS DE CUENTA DE LOS USUARIOS.....144

Glosario

WI-P.O.S: Término usado para denominar el dispositivo realizado en el proyecto. Sus siglas significan *Wireless Point of Sales*. Punto de Ventas Inalámbrico.

P.O.S: *Point of sales*, Punto de ventas

iDEN: (*integrated Digital Enhanced Network*), Red desarrollada por Motorola e implementada en Colombia por AVANTEL, que usa una tecnología permite ofrecer comunicación de voz y transmisión de datos en un sólo dispositivo.

AS400: Servidor desarrollado por IBM, comúnmente utilizado en la administración de procesos bancarios.

DES: (*Data Encryption Standard*) es un sistema estándar de uso internacional para convertir datos en secuencias de bits, que carecen de significado, cuando se encuentran en tránsito por el medio de comunicación.

CBC: (*Cipher Block Chaining Mode),* en DES es un mecanismo de realimentación, donde el próximo dato a cifrar se le hace una XOR con el dato anteriormente cifrado. El primero de los datos a cifrar se le hace una XOR con un vector de inicio.

J2ME: (*Java 2 Platform, Micro Edition*) es una plataforma para la nueva generación de aplicaciones inalámbricas desarrollada especialmente para dispositivos con recursos reducidos de memoria y procesamiento, tales como teléfonos móviles y asistentes digitales personales (PDA's).

PIN: (*Personal Identification Number*), es el número de identificación personal para una tarjeta, en otras palabras es la clave.

PAN: (*Personal Account Number*), es el número de la cuenta del usuario de la tarjeta.

FPGA: (*Field Programmable Gate Array*), es un arreglo de compuertas lógicas que se puede programar con herramientas como ALTERA.

PINBLOCK: El PINBLOCK es una trama de 16 caracteres que se construye haciendo operaciones lógicas entre el PIN y el PAN (Número de cuenta). Este se introduce en la trama debidamente cifrado con la llave de cifrado de PIN. Este espacio de la trama se llena dependiendo si se debe o no solicitar el PIN; si no, se deja vacío.

MSSP *(Master Synchronous Serial Port*), Puerto Maestro Serial Sincrónico.

AHDL: Altera Hardware Description Language.

Consecutivo: Número único por transacción realizado en un mismo terminal P.O.S.

Número de terminal: Número que identifica el P.O.S., no se puede repetir.

Vector de inicio: Son 64 bits con los que se hace una operación lógica XOR del primer bloque de 64 bits a cifrar usando el modo CBC de DES.

1. INTRODUCCION

En los últimos años, el constante crecimiento de la economía, la apertura a nuevos mercados y los avances tecnológicos han abierto las puertas a nuevas formas de comercio. Con ánimo de expandirse, las empresas buscan día a día nuevas estrategias de mercadeo con las cuales atraer la mayor cantidad de clientes posible. La calidad de los productos y el servicio al cliente son factores esenciales para lograr la fidelidad de los compradores potenciales y es por eso que se desarrollan nuevas y mejores soluciones que llaman la atención del consumidor.

Un aspecto muy importante para el consumidor es el tener diferentes alternativas de pago. Este factor hace que la gente utilice cada vez más las tarjetas de crédito y débito, ya que aparte de ser mas cómodo, ofrece ventajas adicionales.

Conscientes de esta realidad, son cada vez más los establecimientos comerciales que permiten a sus clientes el pago de sus bienes y/o servicios con tarjetas de crédito o débito. Para la realización de estos pagos se utilizan unos dispositivos electrónicos llamados datáfonos que se encargan de comunicarse con la entidad bancaria para validar la transacción. La importancia en el comercio de este tipo de pagos han hecho que el desarrollo y mejoramiento de estos dispositivos esté en crecimiento.

En Colombia existen varias empresas cuyo negocio es, entre otros servicios, el de proveer medios electrónicos de pago a los establecimientos comerciales. Uno de los medios de pago que requiere mayor inversión es el de los datáfonos, porque cada dispositivo debe ser adquirido por la empresa y colocado en el establecimiento comercial para que éste pueda realizar transacciones, que son la fuente de ingresos más importante dentro de este modelo de negocios. En los establecimientos, los datáfonos generalmente están conectados a la línea telefónica, sin la cual no podrían acceder a la red de la entidad que cursa las transacciones. La necesidad de la línea

telefónica hace que el datáfono no sea portátil. Además, cada establecimiento debe cubrir los costos de las transacciones y de la factura de la línea telefónica.

Para solucionar el inconveniente de tener datáfonos fijos se han desarrollado algunos equipos inalámbricos. Algunos de estos equipos permiten pagos en lugares remotos al establecimiento, lo que permite mayores alcances y penetración en el mercado ya que el comerciante llega directamente a donde está el cliente. En estos momentos estos dispositivos son todavía muy costosos y no están al alcance de todo el comercio.

Este proyecto pretende satisfacer una necesidad del mercado, mediante el desarrollo de un dispositivo portátil, que puede conectarse a una red inalámbrica por medio de un teléfono móvil, para la realización de pagos a través de tarjetas débito o crédito con tecnología de banda magnética. Todo esto está dirigido desde la pantalla del equipo móvil en un ambiente agradable y sencillo. El dispositivo además asegura la integridad y confidencialidad de la información. El hecho de utilizar un teléfono móvil como puente de comunicación a la red inalámbrica hace que el dispositivo desarrollado en este Trabajo de Grado pueda ser más económico y además proporcionar como valor agregado un teléfono móvil completamente funcional que le además le provea comunicación al usuario final.

Para el desarrollo de este proyecto, la red de datos que se utiliza es la red iDEN, desarrollada por Motorola e implementada en Colombia por la compañía Avantel S.A. la cual ofrece diferentes tipos de servicios dentro de los cuales están: comunicación vía radio, acceso telefónico, mensajes de texto y transmisión de datos.

2. OBJETIVOS

Objetivo General: Desarrollar un dispositivo electrónico que, conectado a un teléfono móvil iDEN, sea capaz de realizar una transacción bancaria segura.

Objetivos específicos:

- Implementar una aplicación en J2ME y un dispositivo electrónico compatible con el equipo Motorola iDEN modelo i85s o i58sr, para realizar una transacción bancaria con tarjetas débito y crédito.
- Desarrollar una aplicación en J2ME para interactuar con el dispositivo electrónico a través del puerto serial del teléfono móvil Motorola.
- Aplicar la correspondiente mensajería entre el teléfono móvil Motorola y el servidor encargado de aceptar las transacciones bancarias para el éxito de ésta, asegurando la confidencialidad y la integridad de la información.

3. ESPECIFICACIONES DEL DISPOSITIVO

El dispositivo, que de ahora en adelante también denominaremos WI-P.O.S (*Wireless Point of Sales*, Punto de ventas Inalámbrico), consiste en un desarrollo que involucra la comunicación del dispositivo con un servidor, programación de un teléfono móvil MOTOROLA, manejo de periféricos necesarios para la realización de una transacción bancaria y cifrado de la información para asegurar la confidencialidad de ésta. Todo esto es realizado bajo diferentes esquemas de implementación, utilizando desarrollos en Hardware, Software y Firmware.

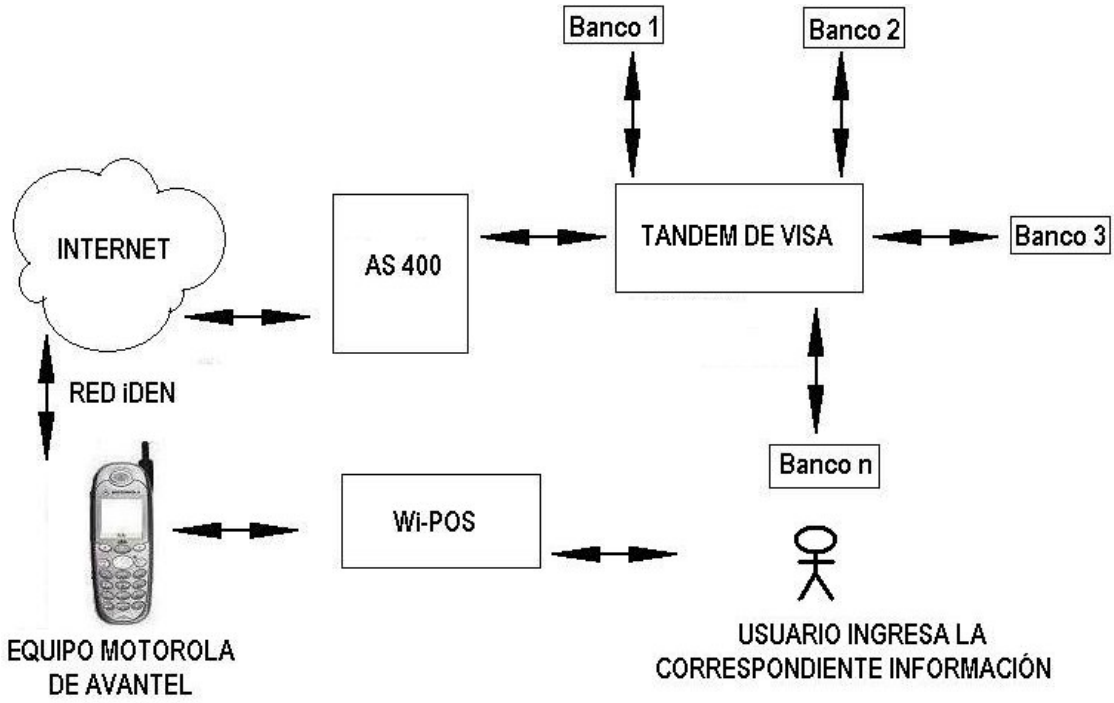


FIGURA 1. ESQUEMA GENERAL DE TRANSACCIÓN CON SERVIDOR DE VISA

En la Figura 1 se puede observar un esquema sencillo para la realización de una transacción bancaria a partir de WI-P.O.S. El equipo recibe la información ingresada por el usuario, que depende del tipo de transacción que vaya a realizar, y construye

las tramas necesarias que luego envía al teléfono móvil para que éste se comunique a través de la red iDEN (ver Anexo Marco Teórico) con un servidor llamado AS400 (es un servidor con tarjeta criptográfica, esta es una tarjeta capaz de realizar funciones de cifrado, entre las cuales se encuentra la posibilidad de cifrar con el algoritmo DES)) que después se comunicará con el Tandem[1] de VISA y esperará respuesta de la aprobación de la transacción. Este proceso se realiza por que el protocolo que maneja las transacciones (Base 24 ver Anexo Marco Teórico) es muy robusto y sería necesaria una gran cantidad de procesamiento en el equipo que incrementaría en gran medida los costos y la complejidad del desarrollo. De manera que existe una aplicación desarrollada por una empresa colombiana (ASIC)[2] que recibe cierta información que es recopilada en el P.O.S. (*Point of sales*, Punto de Ventas) y haciendo uso de esta construye las tramas correspondientes al protocolo Base24 (Ver sección 3.3). En la siguiente sección se presentará un esquema detallado de la realización de una transacción de pago ya sea con tarjeta débito o crédito.

3.1. DIAGRAMAS DE FLUJO DE LOS PROCESOS NECESARIOS PARA LLEVAR A CABO UNA TRANSACCIÓN DE PAGO CON TARJETA CRÉDITO O DÉBITO USANDO WI-P.O.S.

3.1.1. Diagrama de proceso para guardar llaves de cifrado y terminal.

Por cuestiones de seguridad toda la información introducida por el usuario al dispositivo es cifrada usando el algoritmo *DES* (*Data Encryption Standard*) (ver Anexo Marco Teórico). Por lo tanto, previo a la realización de cualquier transacción, WI-P.O.S debe tener almacenadas una llave maestra y una llave de transporte (Ver Sección 3.4). Las llaves se cargan conectando WI-P.O.S al puerto serial de un computador y por medio de una aplicación realizada en Visual Basic, estas se guardan en la memoria EEPROM del dispositivo. La llave maestra se usa para cifrar las otras llaves y almacenarlas de manera segura. La llave de transporte se usa para

[1] Tandem es un nodo en donde se conectan los servidores de los Bancos, y es en estos donde se valida la transacción.
[2] ASIC, Empresa desarrolladora de aplicaciones en hardware y software, orientada al sector financiero.

cifrar cualquier información que se transmita al servidor durante la transacción.

Cada dispositivo debe tener un número que lo identifique, es por esto que también se debe cargar un número de terminal. El número de terminal es representado por 8 caracteres.

A continuación se ilustra el diagrama de flujo de la aplicación realizada en Visual Basic:

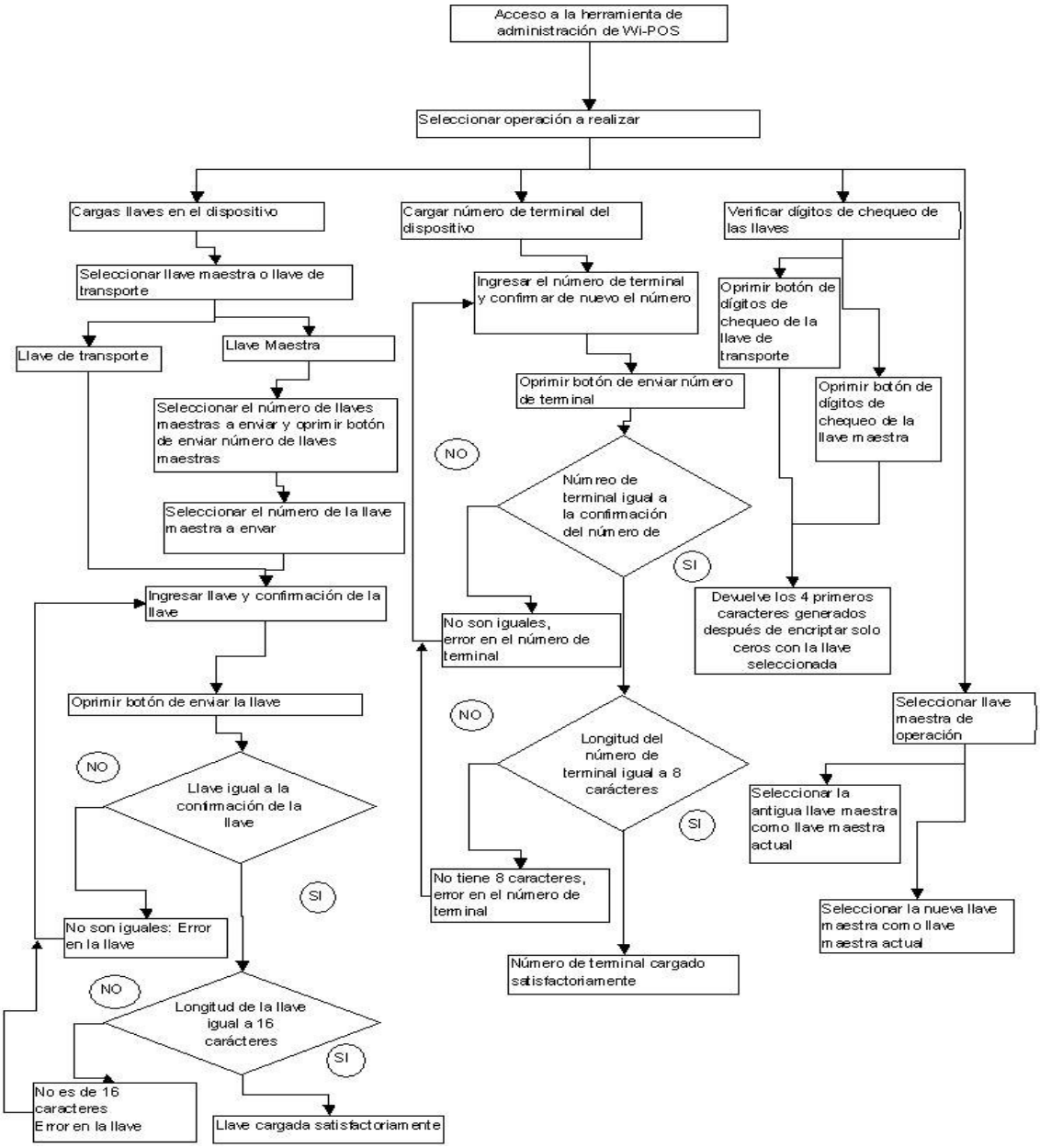


DIAGRAMA DE FLUJO 1
Aplicación de Visual Basic, Administrador de Herramientas para WI-P.O.S Parte 1

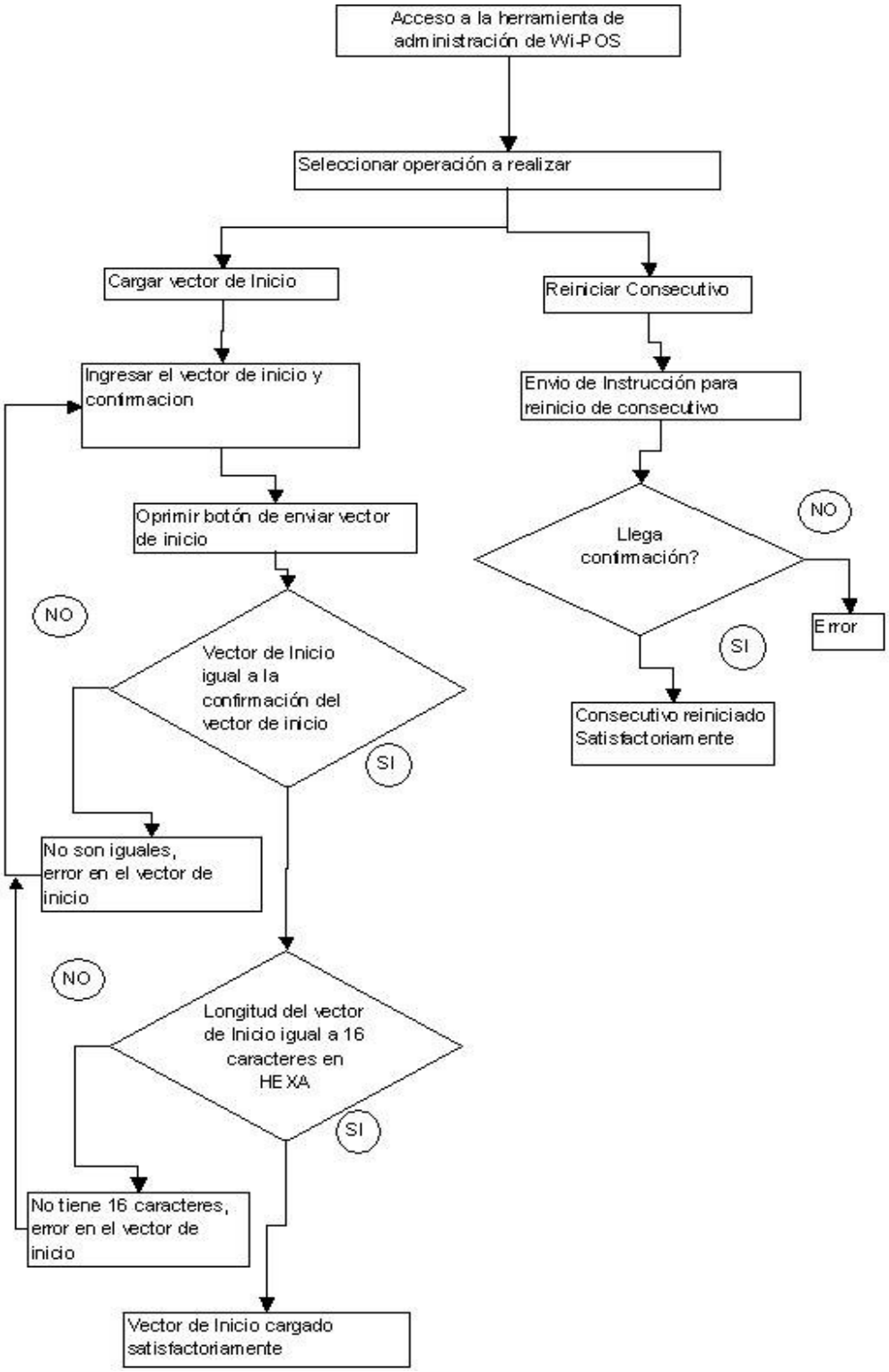


DIAGRAMA DE FLUJO 2
Aplicación de Visual Basic, Administrador de Herramientas para WI-P.O.S Parte 2

Esta aplicación realiza 5 funciones principales que son cargar las llaves en el dispositivo, cargar su correspondiente número de terminal, y realizar la operación de verificación de dígitos de chequeo.

i) Cargar las llaves: Es un proceso que permite introducir las llaves maestras o la llave de transporte al dispositivo desde el computador. Para evitar que se ingresen valores erróneos el proceso es redundante, es decir, se ingresa dos veces la información de las llaves, además se verifican otros datos como el tamaño de la llave, que tiene que ser exactamente de 16 caracteres, tomando valores hexadecimales (0-F).

ii) Cargar número de terminal: Es un proceso que sirve para introducir el número de terminal del dispositivo. Este valor debe ser único por cada uno, ya que es quien lo identifica. Al igual que el proceso anterior, la información se debe ingresar dos veces para evitar que se introduzcan valores erróneos y su tamaño debe ser de 8 dígitos.

iii) Dígitos de chequeo: Es un proceso necesario para saber si las llaves ingresadas son correctas. Consiste en el cifrado de ceros con la llave seleccionada, ya sea maestra o transporte, de esta forma la aplicación muestra los 4 primeros valores de la operación y se verifica si son correctos.

iv) Cargar Vector de Inicio: Este proceso consiste en cargar el vector de inicio en el dispositivo, tiene una longitud de 16 caracteres en hexadecimal, este valor es indispensable para el correcto cifrado de la información en modo CBC, proceso que será explicado en los siguientes capítulos.

v) Reinicio de consecutivo: Es un proceso por el cual a un dispositivo Wi-P.O.S. se le asigna el valor de consecutivo en ceros, que identifica las transacciones.

3.1.2 Diagrama de flujo de carga de parámetros iniciales

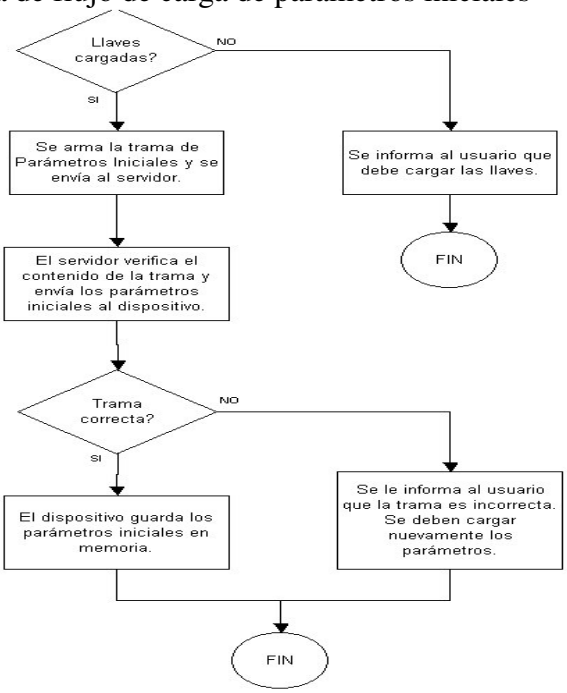


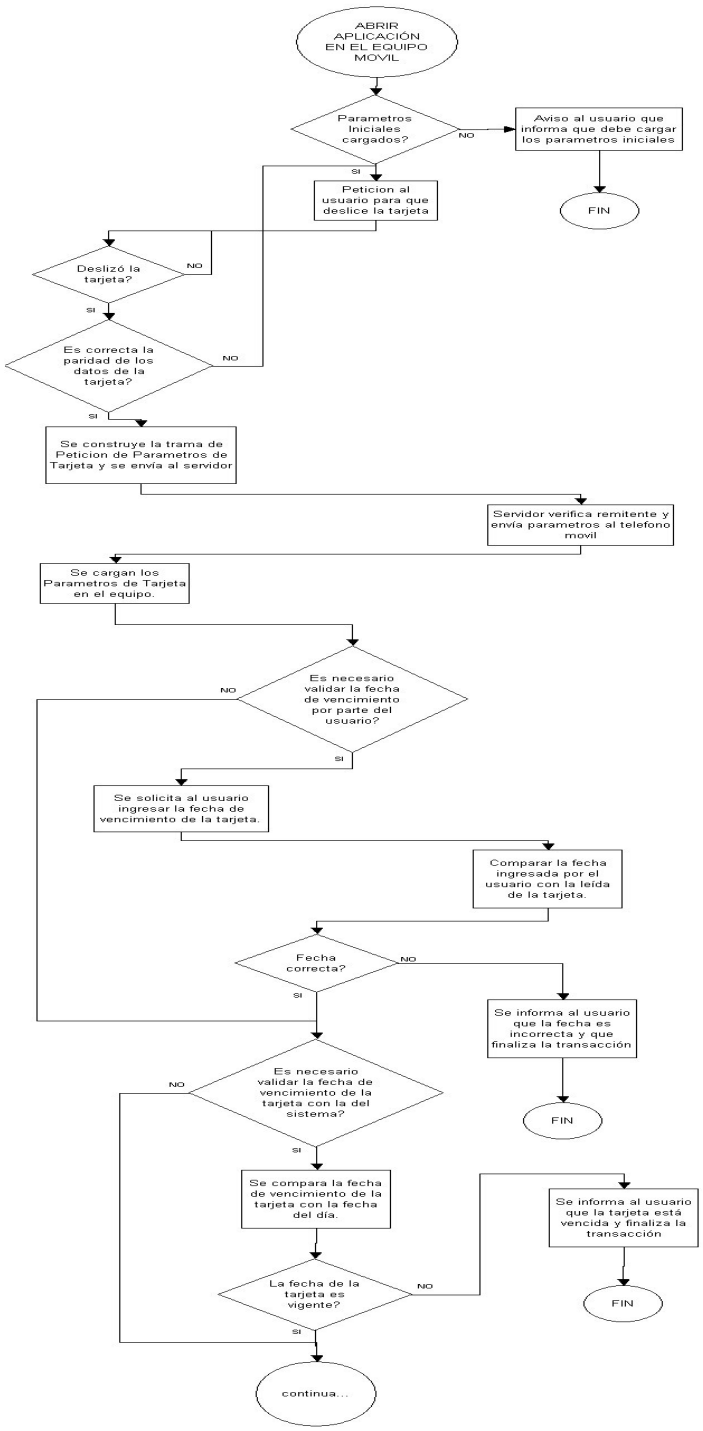
Diagrama de flujo 3. Petición de parámetros iniciales

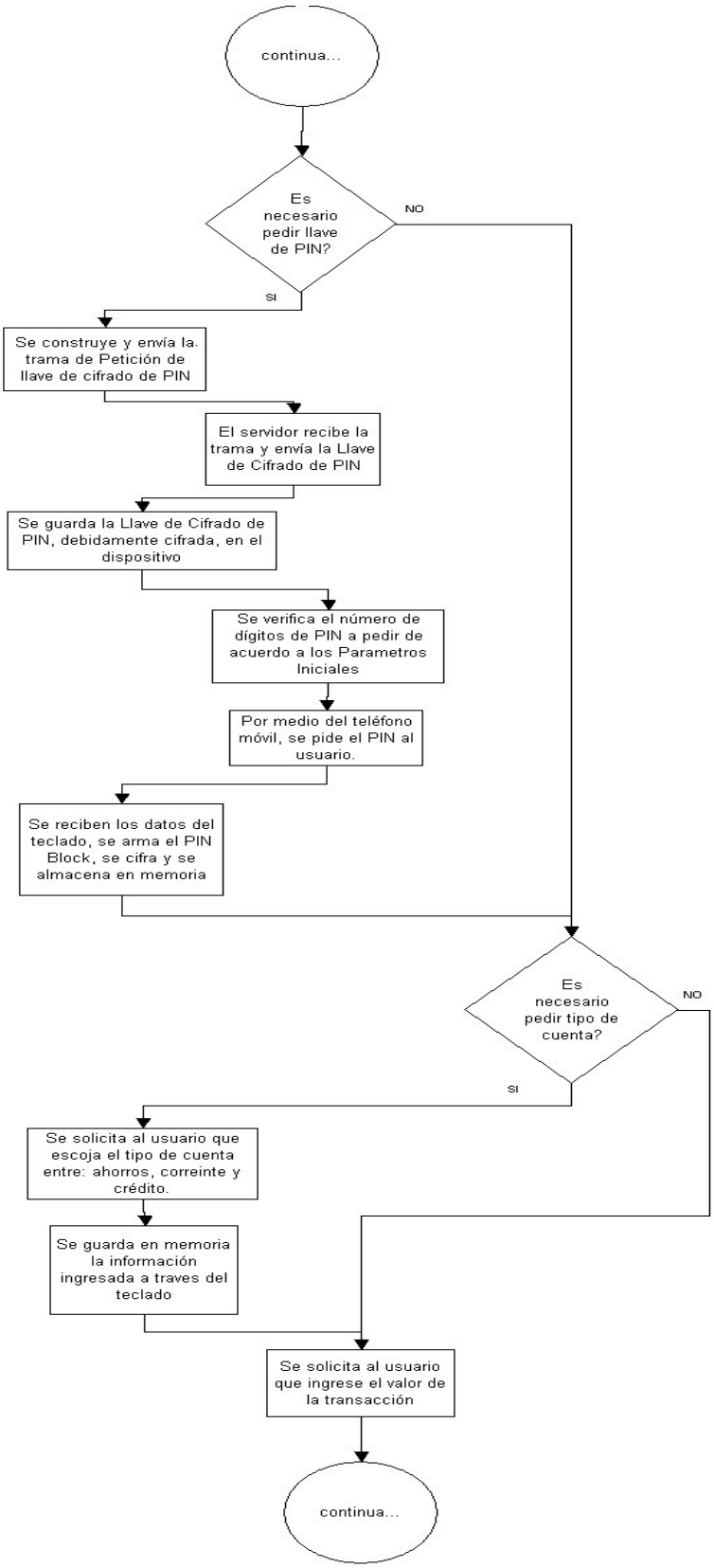
Además de tener las llaves almacenadas, WI-P.O.S debe haber cargado (guardado en memoria) unos parámetros iniciales que obtiene de una conexión con el servidor. Esto último se debe hacer cada vez que se encienda el equipo. En caso de que alguno de estos pasos no se haya realizado, es decir si no se han cargado las llaves (de transporte y maestra) y/o no se han cargado los parámetros iniciales, no se podrá llevar a cabo la transacción. Por esto cada vez que se va a realizar una transacción el equipo verifica que estos datos estén almacenados en el dispositivo y envía al teléfono un mensaje de confirmación.

Para cargar los parámetros iniciales se debe correr una aplicación en J2ME cada vez que se enciende el equipo, de manera que WI-P.O.S arma la trama denominada "petición de parámetros inicial" (Ver sección 3.3) y la cifra usando la llave de transporte para enviarla al servidor. Este devuelve una trama de parámetros iniciales los cuales se descifran y se almacenan en la memoria del dispositivo. Estos parámetros iniciales incluyen el porcentaje de IVA, solicitud de propina, dígitos a ingresar (para la clave) y el nombre del emisor. Estos dependen del terminal que hace

la petición. (Ver sección 3.3).

3.1.3 Diagrama de flujo de la realización de una transacción de pago con tarjeta edito o débito:




continua...
Es necesario pedir llave de PIN?
NO
SI
Se construye y envía la trama de Petición de llave de cifrado de PIN
El servidor recibe la trama y envía la Llave de Cifrado de PIN
Se guarda la Llave de Cifrado de PIN, debidamente cifrada, en el dispositivo
Se verifica el número de dígitos de PIN a pedir de acuerdo a los Parametros Iniciales
Por medio del teléfono móvil, se pide el PIN al usuario.
Se reciben los datos del teclado, se arma el PIN Block, se cifra y se almacena en memoria
Es necesario pedir tipo de cuenta?
NO
SI
Se solicita al usuario que escoja el tipo de cuenta entre: ahorros, correinte y crédito.
Se guarda en memoria la información ingresada a traves del teclado
Se solicita al usuario que ingrese el valor de la transacción
continua...

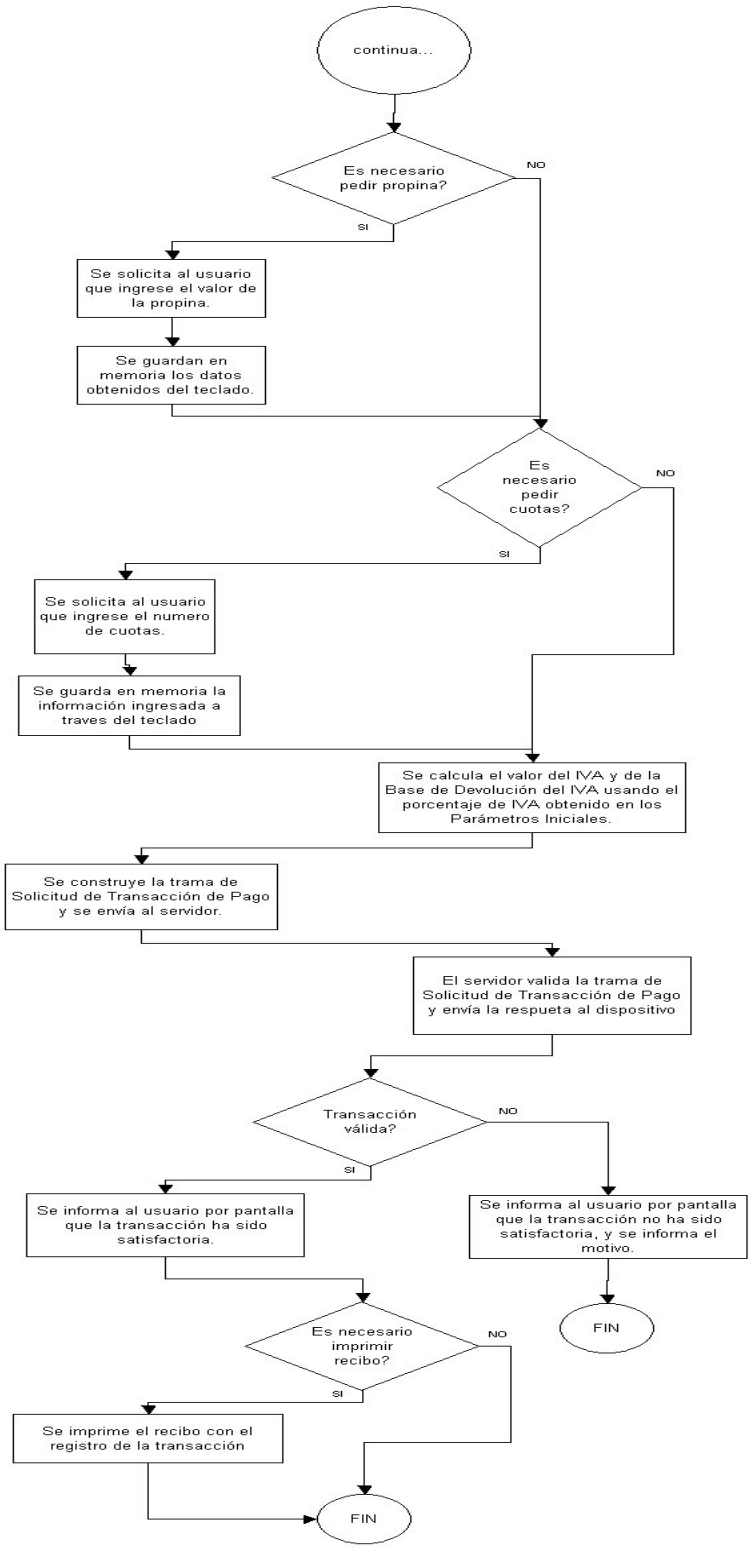


DIAGRAMA DE FLUJO 4
Transacción de Pago con tarjeta débito o crédito

Después de verificar que los parámetros iniciales estén cargados, por medio de la pantalla del teléfono se le pide al usuario deslizar la tarjeta. WI-P.O.S recibe la información de la tarjeta y arma una trama denominada "petición de parámetros de tarjeta". Esta trama se cifra y por medio del teléfono móvil se envía al servidor, el cual devuelve los parámetros de la tarjeta (Ver sección 3.3) que se almacenan en el dispositivo. Después de esto se verifica con los parámetros de tarjeta recibidos del servidor si es necesario el ingreso de PIN (*Personal Identification Number*) o clave personal, y de serlo así, arma una trama denominada "petición de llave de cifrado de PIN" la cifra y la envía al servidor el cual devuelve la llave que luego se almacena en WI-P.O.S. Esta llave sirve para cifrar el *PIN Block* (Ver sección 3.3) que es una trama que se construye haciendo uso del PIN y del PAN (*Personal Account Number*, Número de Cuenta). Además de ser necesario, el equipo deberá verificar la fecha de vencimiento de la tarjeta, y compararla con la fecha ingresada por el usuario al equipo, o con la fecha del sistema.

Luego el dispositivo, según los parámetros de tarjeta recibidos, pide al usuario los campos correspondientes (valor, cuotas, tipo de cuenta entre otros), arma una trama denominada "pago con tarjeta", la cifra, la envía al servidor, y espera respuesta.

Cabe anotar que durante el proceso de la transacción, todas las tramas enviadas y recibidas del servidor van debidamente cifradas usando el algoritmo *DES* para garantizar la seguridad y confidencialidad de la información que se transmite en ambas direcciones, desde el teléfono y hacia el teléfono.

WI-P.O.S también es capaz de generar transacciones de reverso (ver sección 3.3), esto en caso que nunca se encuentre respuesta o que se genere algún error durante la realización de la transacción. Por ejemplo si en el momento de realizar una transacción ésta fue aceptada pero el mensaje de respuesta no llegó nunca al teléfono móvil. Al no obtener respuesta en el P.O.S. del resultado de la transacción se debe generar un reverso que anulará la correspondiente transacción. (Ver sección 3.3).

3.2 ESPECIFICACIONES DE WI-P.O.S.

WI-P.O.S es un dispositivo que posee los elementos necesarios para la realización de una transacción de pago con tarjeta débito o crédito. Estos elementos son: un lector de tarjetas de banda magnética, un teclado (para el ingreso del valor de la transacción, *PIN*, número de cuotas, etc.), un módulo de cifrado *DES,* un puerto serial y una salida para impresora. El dispositivo tiene una comunicación constante con el teléfono móvil, quien es el que le envía las instrucciones según la implementación de su programa a partir de los módulos y clases en J2ME, desarrollados como parte del trabajo. Además de guiar la transacción, el teléfono móvil tiene el muy importante papel de enviar la información al servidor.

3.2.1 Entradas y salidas del dispositivo

Entradas del dispositivo

- Lector de Banda magnética
- Teclado de 4 Filas x 3 Columnas
- Entrada serial usando el estándar RS-232
    - Conector DB9.

Salidas del dispositivo

- Salida serial por estándar RS-232 conector DB9
- Salida conector DB9 para impresora.

A continuación se mostrará un esquema de las entradas y salidas de WI-P.O.S:

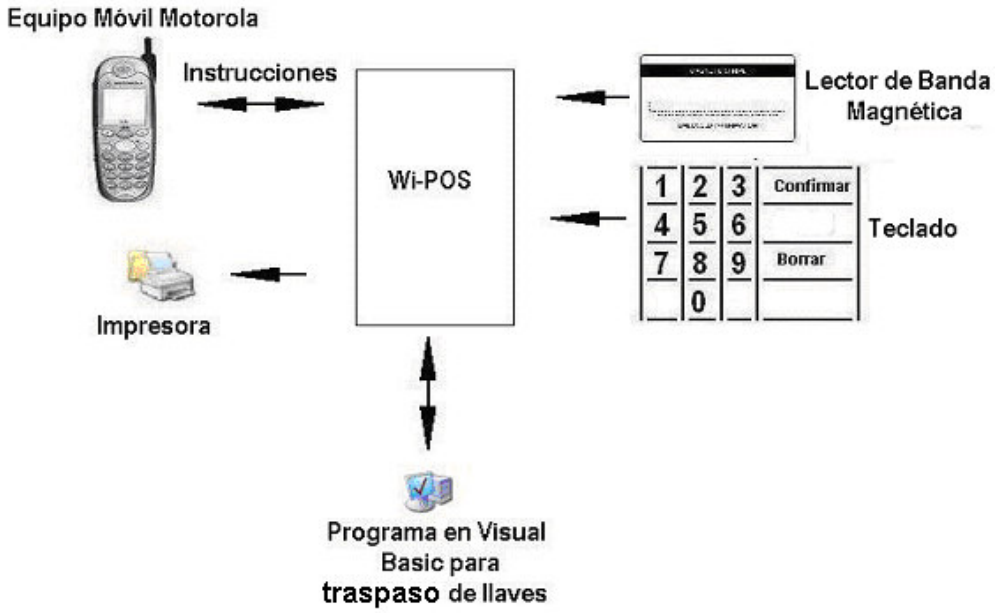


FIGURA 2. ESQUEMA DE ENTRADAS Y SALIDAS DE WI-P.O.S

En la Figura 2 se pueden observar las diferentes entradas y salidas que tiene WI-P.O.S, por medio de las cuales se establece la comunicación, explicada en las siguientes secciones

- COMUNICACIÓN CON EL USUARIO: El usuario sólo puede interactuar con WI-P.O.S deslizando la tarjeta de banda magnética por el correspondiente lector y digitando la información solicitada según la instrucción desplegada en la pantalla del teléfono. WI-P.O.S tiene su propio teclado debido a que de esta forma la información digitada no pasa directamente por el teléfono móvil y no se corre el riesgo que se pueda almacenar en éste sino que toda la información se cifra y se guarda directamente en WI-P.O.S, evitando problemas de retención de información, haciendo que la transacción sea más segura. El usuario también tiene derecho a adquirir el comprobante de su transacción bancaria por medio de una pequeña impresora que se conecta al equipo.

- COMUNICACIÓN CON TELÉFONO MÓVIL: El teléfono móvil se conecta al puerto serial de WI-P.O.S y es quien "dirige" la transacción, ya que él es quien envía la instrucción correspondiente a la operación desplegada en

pantalla. Por ejemplo, en el momento que se solicite digitar la clave, el teléfono mostrará una pantalla para que el usuario la digite, al mismo tiempo enviará la instrucción a WI-P.O.S para que se prepare a recibir la correspondiente información del teclado y luego manipularla. Es decir que para cada una de las fases de la transacción, el teléfono móvil muestra en la pantalla lo que el usuario debe realizar y a la vez enviará las instrucciones a WI-P.O.S para que éste entre a la rutina correspondiente.

- INGRESO DE LLAVES: Para el ingreso de las llaves Maestra y de Transporte se debe conectar el WI-P.O.S con el puerto serial de un computador y haciendo uso de la aplicación desarrollada en Visual Basic almacenar estas en la memoria EEPROM del microcontrolador, debidamente cifradas.

3.2.2 Componentes del dispositivo

- Microcontrolador PIC18F452
- Integrado MAX 232
- Integrado 74C422N (Manejador del teclado)
- Integrado 74LS125
- Integrado LM555
- Cristal de 4.9152 MHZ
- FPGA Altera ACEX1K50TI144-2
- Memoria Serial Altera EPC2LC20
- Oscilador con cristal de 1MHz
- Regulador LM1117DT-2.5
- Regulador LM1117DT-3.3
- Resistores
- Condensadores
- 2 leds
- Interruptor
- Lector de Tarjetas de banda magnética Omron V3A

- Teclado de 4filas x 3columnas

3.2.3. Módulos del dispositivo:

El dispositivo cuenta con tres módulos que se listan a continuación, luego serán explicados.

- Modulo de manejo de periféricos
- Modulo de cifrado en DES
- Desarrollo de Software. Aplicaciones en J2ME y Visual Basic.

- Modulo de manejo de periféricos

El módulo de manejo de periféricos consiste de un microcontrolador PIC 18F452 que se encarga de manejar la comunicación, con el teléfono móvil, con la impresora, con el módulo de cifrado y además recoge la información del lector de banda magnética y del teclado. Este se encarga de almacenar toda la información, y de armar las tramas correspondientes a la transacción.

- Módulo de cifrado en DES

Este es un cifrador/descifrador en hardware (implementado en una FPGA) que permite manejar información de una manera confiable. Este trabaja con el estándar de cifrado DES (ver Anexo Marco Teórico) que utiliza una llave de 64 bits. Este realiza el cifrado/descifrado en bloques de 64 bits. La información se recibe y se envía serialmente usando una interfase de puerto serial y trabajando como esclavo. Es decir que sólo envía la información cuando le es solicitada por otro. Tiene un bit de selección de funcionamiento (cifrar/descifrar).

- Desarrollo de Software

El teléfono móvil es la interfaz visual con el usuario para la realización de la transacción, por tanto éste establece una comunicación con WI-P.O.S indicándole los pasos a seguir, además informa al usuario las acciones que debe realizar. Por esto fue necesario desarrollar tres aplicaciones: la primera para la petición de los parámetros

iniciales, la segunda para la realización de un pago y la tercera para reversar una transacción. Este es un programa (Software) que denominamos “driver” por que es el que maneja el dispositivo; éste programa incluye todos los pasos y pantallas necesarias para la transacción. Este Software debe ser instalado en el teléfono móvil antes de usar WI-P.O.S, por tanto es necesario que el teléfono móvil a usar pueda ser programado usando J2ME. Además se desarrolló una aplicación en Visual Basic que permite introducir las llaves y el número de terminal, para que sean almacenadas en WI-P.O.S.

3.3 PROTOCOLO DE MENSAJES PARA REALIZACIÓN DE LA TRANSACCIÓN.

El desarrollo realizado usa como puente un servidor para la realización de la transacción, este tiene una aplicación implementada por ASIC, y se encarga de recibir ciertos datos del P.O.S y de construir las tramas correspondientes al protocolo de Base 24 (ver Anexo Marco Teórico). Esto permite que el procesamiento de datos se reduzca en el punto de ventas ya que el protocolo de Base 24 es bastante robusto y maneja tramas grandes y complejas. De manera que el equipo desarrollado (WI-P.O.S) tiene comunicación directa con el servidor AS400 y este último es quien se encarga de conectarse con el Tandem de VISA.

A continuación se mencionarán los mensajes que se deben construir en el equipo, para luego enviarlos al servidor AS400 y así poder llevar a buen término la transacción. Todos estos mensajes se transmiten debidamente cifrados con la llave de transporte. Los primeros dos bytes de todos los mensajes se denominan tipo de mensaje y son estos los que identifican el mensaje que se envía o que se recibe. A continuación se presentarán los mensajes, con una breve descripción y los campos que los componen.

3.3.1. Solicitud de parámetros iniciales:

Esta trama se debe enviar al servidor cada vez que se prende el equipo, con el fin de obtener del servidor una serie de parámetros que dependen del tipo de terminal, del propietario del equipo y de la zona de uso.

La trama tiene los siguientes campos:

i) Tipo de mensaje: Dos bytes que identifican el mensaje, estos son de tipo char.

ii) Terminal: Es un código que identifica cada uno de los equipos (datáfonos), éste consiste de 8 bytes de tipo char.

3.3.2. Respuesta de parámetros iniciales:

ésta respuesta debe ser enviada por el servidor inmediatamente después de recibir la trama de petición de parámetros iniciales. Consiste de los siguientes campos.

i) Tipo de mensaje: 2 bytes de tipo **char**

ii) Terminal: Debe transmitir el número de terminal del equipo que envió la solicitud, de ésta manera se puede comprobar en el P.O.S si el mensaje recibido es correcto.

iii) Porcentaje de IVA: Esta compuesto por dos datos de tipo numérico, cada uno de 1 byte.

iv) Solicitud Propina: es un carácter que indica si se debe o no pedir la propina en el equipo.

v) Numero de dígitos a ingresar: Este campo compuesto por dos datos numéricos indica la cantidad de dígitos que se deben ingresar al pedir el PIN.

vi) Base devolución de IVA: es un carácter que indica si debe o no viajar la base de devolución de IVA en el momento de hacer la petición de transacción de pago.

vii) Nombre del Emisor: Es un campo de 10 caracteres que contiene el nombre de la entidad que presta el servicio.(ej: Visa)

3.3.3. Solicitud de parámetros de tarjeta:

Esta trama se transmite al servidor después de que se ha deslizado correctamente la tarjeta en el lector. Se hace con el fin de recibir una serie de parámetros que dependen de la tarjeta deslizada. Sus campos son:

i) Tipo de Mensaje.

ii) Número de terminal.

iii) Número de la tarjeta: Este campo incluye los datos obtenidos de la Pista 2 de la banda magnética de la tarjeta (Ver Anexo Marco Teórico). Son 19 bytes de tipo **char**.

3.3.4. Respuesta de parámetros de tarjeta:

Es la trama que transmite el servidor como respuesta a la petición de parámetros de tarjeta y que incluye los siguientes campos:

i) Tipo de Mensaje.

ii) Número de Terminal

iii) Número de la tarjeta: Se transmite de vuelta el número de la tarjeta para corroborar que los parámetros si correspondan a esa tarjeta.

iv) Validación fecha de vencimiento1: Es un carácter (de 1 byte) que indica si se debe o no solicitar al usuario que ingrese la fecha de vencimiento de la tarjeta, para corroborar con lo leído en la PISTA2.

v) PIN para tarjetas de crédito: Es un carácter que indica si se debe o no solicitar PIN con esa tarjeta de crédito.

vi) Solicitud de tipo de cuenta: Es un carácter que indica si se debe solicitar al usuario el tipo de cuenta.

vii) Validación fecha de vencimiento: Es un carácter que indica si se debe validar localmente o no la fecha de vencimiento obtenida de la PISTA2 con la fecha del sistema.

viii) Impresión Recibo: Este carácter indica si se debe o no imprimir el recibo de Credibanco.

ix) Módulo 10: Un carácter que indica si se le debe hacer la validación de módulo 10 al número de cuenta de la tarjeta (ver Anexo Marco Teórico)

x) Solicitud de cuotas: Es un carácter que indica si se debe o no solicitar un número de cuotas

xi) Solicitud de PIN: Este carácter indica si se debe pedir PIN para tarjetas débito.

3.3.5 Solicitud de transacción de pago:

Es la trama que se debe construir después de haber solicitado todos los datos necesarios. Contiene los siguientes campos:

i) Tipo de Mensaje

ii) Número de terminal

iii) Consecutivo: Este es un número compuesto por seis datos de tipo numérico (cada uno de un Byte) que identifica cada transacción. Este número se debe aumentar cada vez que se realiza una transacción. Este dato se encuentra almacenado en memoria EEPROM.

iv) PISTAII: Se debe enviar toda la información del PISTAII de la tarjeta deslizada. Tiene una longitud de 37 caracteres (cada uno representado por un Byte)

v) PISTAI: Si la tarjeta tiene información en el PISTAI este se debe enviar, tiene una longitud de 76 caracteres.

vi) PINBLOCK: El PINBLOCK es una trama de 16 caracteres que se construye haciendo operaciones lógicas entre el PIN y el PAN (Número de cuenta). Este se introduce en la trama debidamente cifrado con la llave de cifrado de PIN. Este espacio de la trama se llena dependiendo si se debe o no solicitar el PIN; si no, se deja vacío.

vii) Tipo de cuenta: Un dato numérico (representado en un Byte) indica el tipo de cuenta del usuario. Este campo se llena o no dependiendo de los parámetros de la tarjeta.

viii) Cuotas: 2 bytes indican el número de cuotas. Viaja o no dependiendo de los parámetros.

ix) Valor de la transacción: Es el valor por el cual se realiza la transacción, se representa en doce datos de tipo numérico en donde los últimos dos se usan para centavos.

x) IVA: Es la parte del valor de la transacción que corresponde al IVA se representa con doce datos numéricos.

xi) Base de devolución de IVA: Se representa con doce datos numéricos, viaja o no dependiendo de los parámetros iniciales.

3.3.6. Respuesta de transacción de pago

Tiene los siguientes campos

i) Tipo de Mensaje

ii) Número de terminal.

iii) Consecutivo: Se envía para saber si la respuesta si corresponde a la debida transacción.

iv) Código de respuesta: Son dos caracteres (cada uno representado en 1 Byte) que indican si la transacción fue exitosa o si existe algún error.

v) Descripción de respuesta: 20 caracteres que describen la aprobación o el error de la transacción.

vi) Código de autorización: Es un código de 6 datos de tipo numérico, diferentes de cero cuando la transacción es aprobada.

3.3.7. Petición de transacción de reverso:

Es una trama que se construye en caso de necesitar un reverso. El reverso anula o revierte la transacción indicada por el consecutivo enviado. Los campos son:

i) Tipo de Mensaje.
ii) Número de terminal.
iii) Consecutivo

3.3.8. Respuesta de transacción de reverso:

Tiene los mismos campos que la respuesta de transacción de pago pero se identifica con otro tipo de mensaje.

3.3.9. Solicitud de llave de cifrado de PIN:

Con esta trama se hace una solicitud al servidor de una nueva llave para el cifrado de PIN, contiene los siguientes campos:

i) Tipo de Mensaje.

ii) Número de terminal.

3.3.10. Respuesta de llave de cifrado de PIN:

Contiene los siguientes campos

i) Tipo de Mensaje.

ii) Número de terminal.

iii) Llave de cifrado de PIN: Contiene 16 caracteres que representan la llave de cifrado de PIN

3.4. MANEJO DE LLAVES DE CIFRADO

3.4.1. Manejo de Llave Maestra**.**

La llave maestra es una llave de cifrado usada para mantener la información de las llaves de transporte y de PIN almacenadas seguramente en el equipo.

El proceso para almacenar una llave maestra en un POS es el siguiente: primero se acuerda cierta cantidad de subllaves que se van a generar, la llave maestra se genera al hacer una operación lógica XOR entre éstas. La finalidad de generar cierta cantidad de subllaves es la de entregarle una a diferentes personas, las cuales ingresarán la información al POS en diferente tiempo, de esta manera nadie conocerá la llave maestra sino solamente un fragmento.

La forma en que se introducen las llaves maestras en un POS es mediante una aplicación que se ejecuta desde un Computador al cual se conecta el POS, ésta conexión depende de las características de cada POS.

En un POS, siempre se deben almacenar máximo 3 llaves maestras, la última llave maestra usada antes de la actualizada, la llave maestra actual y la llave resultante de la XOR de las partes ingresadas. En el momento que se desean actualizar las llaves maestras, éstas sufren un corrimiento, de tal manera que la llave resultante de la XOR pasa a ser la nueva llave actual y la llave actual pasa a ser la última llave.

En caso que se desee volver a usar la última llave como llave maestra actual, se tiene

la opción de hacer que la última llave ocupe de nuevo la actual, quedando vacío el espacio que ocupaba antes. Este proceso se hace en caso que se presenten inconvenientes con la llave maestra actual. Para una mejor idea del proceso se puede ver más detalladamente en la Figura 3.

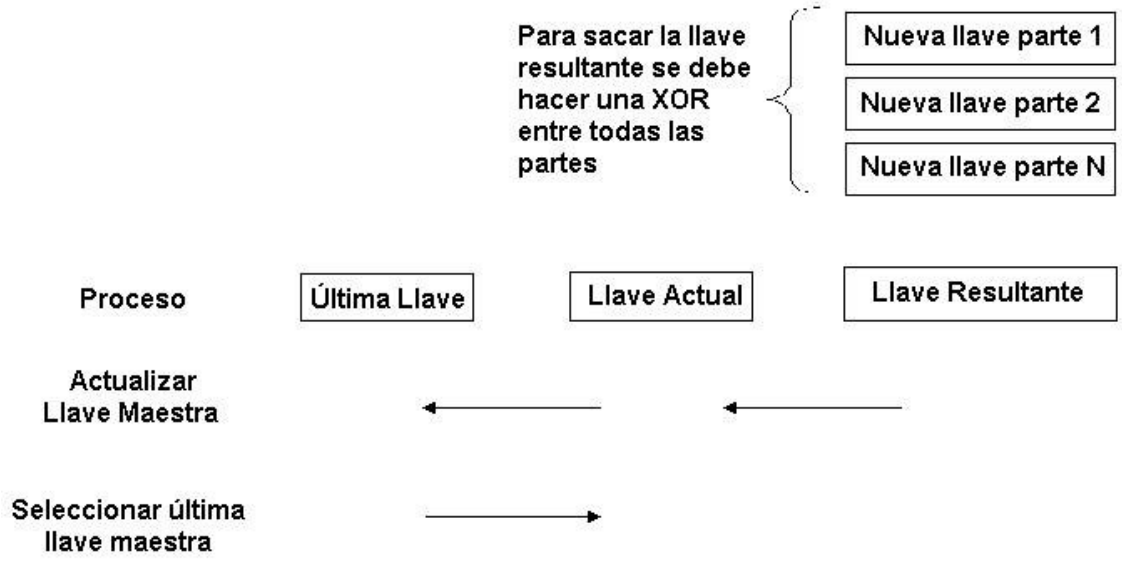


FIGURA 3, Proceso de Carga de Llaves Maestras

3.4.2. Llave de Transporte

Es una llave de cifrado que sirve para enviar y recibir información entre WI-P.O.S. y el servidor, previamente esta llave ha sido concretada entre las dos partes e ingresada en los 2 sistemas. Por seguridad este tipo de llave no permite que tenga un intercambio dinámico.

Por seguridad ésta llave es cifrada dentro de WI-P.O.S por medio de la llave maestra y luego almacenada. Es necesario que sea guardada en una memoria no volátil para evitar que se borre cada vez que se apague el equipo.

Para el ingreso de la llave de transporte se debe hacer mediante una interfaz en un

computador.

3.4.3. Llave de cifrado de PIN

La llave de cifrado de PIN se usa exclusivamente para cifrar el PINBLOCK, esta es una llave que se genera aleatoriamente en la tarjeta criptográfica del servidor y se debe pedir cada vez que se hace una transacción, ésta se almacena en la memoria RAM. La llave de cifrado de PIN se solicita usando la trama denominada petición de llave de cifrado de PIN. (Ver sección 3.3.9)

4. DESARROLLO

El diseño y elaboración del dispositivo denominado WI-P.O.S contó con desarrollos en Hardware, Software y Firmware. Estos fueron divididos en tres áreas principales que serán explicadas a continuación. 1) Manejo de periféricos y procesamiento de datos en el dispositivo, 2) Cifrado de datos y 3) Aplicaciones en J2ME y Visual Basic.

4.1 MANEJO DE PERIFÉRICOS Y PROCESAMIENTO DE DATOS.

Como ya se ilustró en la sección 3.2 los periféricos que maneja el dispositivo son: un lector de banda magnética, un teclado de 4 filas x 3 columnas, una impresora y un teléfono móvil. Además se puede tomar el bloque de cifrado como un periférico más y la comunicación con éste se explica más adelante.

Para manejar los periféricos, como unidad de procesamiento central se eligió un microcontrolador PIC 18F452. Este es un microcontrolador de 8 bits , cuenta con una memoria FLASH de 32Kbytes, una memoria RAM de 1536 bytes y una EEPROM para datos de 256 bytes, cuenta con un USART (Addressable Universal Synchronous Asynchronous Receiver Transmitter) y un módulo MSSP (Master Synchronous Serial Port) éstos últimos muy útiles para el desarrollo. Además tiene 5 puertos de entrada/salida, 1puerto de 7 pines, 3 de 8 pines y uno de 3 pines.

En la rutina principal del microcontrolador, este se encuentra esperando datos a través de la USART, cada vez que ingresa un dato este lo lee y decide a que subrutina debe entrar, ejecuta las acciones correspondientes y vuelve a su estado inicial. Para que el

dispositivo desarrollado contara con versatilidad y posibilidades de expansión, se fijo una serie de rutinas que se realizan dependiendo de la instrucción recibida por el puerto. El dispositivo cuenta con un interruptor de reset para inicializar el equipo en caso de fallas, el circuito de reset se implemento haciendo uso de un integrado LM555.

4.1.1 Comunicación con el teléfono móvil.

La comunicación con el teléfono móvil se hace a través de la USART del microcontrolador. La USART (Addressable Universal Synchronous Asynchronous Receiver Transmitter) es uno de los 2 módulos de transmisión serial con que cuenta el microcontrolador. Este puede funcionar de manera sincrónica o asincrónica. Para el desarrollo del proyecto se decidió trabajarlo de manera asíncrona ya que es así que funciona el teléfono móvil, el cual maneja el protocolo EIA/TIA 232. En modo asíncrono el puerto usa un formato de datos estándar de no retorno a cero. La transmisión se hace de a 8 bits con un bit de inicio. Para la comunicación se eligió una tasa de transmisión de 9600bps. El generador de rata de transmisión divide el reloj de funcionamiento del microcontrolador en un número entero, para el caso específico en que se trabajó al microcontrolador, es decir, en modo de velocidad baja, se usa la siguiente fórmula para la rata de transmisión

$$\text{Tasa de transmisión} = F_{osc}/(64(n+1))$$

En donde n es un número entero que se debe poner en el registro SPBRG (registro de generación de tasa de transmisión de la USART) del microcontrolador al inicializar el puerto y F_{osc} la frecuencia del reloj con que trabaja el microcontrolador. Como n es un número entero, si la frecuencia del reloj no es un múltiplo entero de 9600 se va a generar un porcentaje de error en la transmisión, es por esto que se eligió un reloj de 4.9152Mhz para el microcontrolador, si se remplaza en la fórmula se encontrará un

valor de *n* igual a 7 y un porcentaje de error debido a tasa de transmisión de 0%.

El puerto serial (USART) del microcontrolador se usa para comunicarse con el teléfono, con el computador y con la impresora. Estos tres usan el protocolo EIA/TIA 232 cuyos voltajes exceden el límite de trabajo en el circuito ya que éste opera con una fuente sencilla de 3.3V, que lo hace compatible con la batería del teléfono en caso de querer conectarlo a ésta. Por ésta razón, se hace necesario el uso de un integrado MAX 232 que convierte los voltajes del protocolo EIA/TIA 232 a voltajes compatibles con TTL y CMOS. Es de notar que se cuenta con un sólo puerto (USART) y se necesita establecer comunicación con tres periféricos diferentes. Para solucionar este inconveniente se tuvieron en cuenta los siguientes aspectos: el microcontrolador no se comunica con dos periféricos a la vez, además se conecta con el computador para cargar la llaves y el número de terminal, momento en el cual no debe estar conectado al teléfono móvil, y por último, el integrado MAX 232 cuenta con dos entradas y dos salidas. Como resultado se decidió usar un *buffer tristate* cuyos *output enable* se manejan desde el microcontrolador con dos salidas digitales. De modo que para el teléfono y el computador se usa la misma línea ya que estos no se conectan simultáneamente. La salida Tx del puerto de USART se conecta simultáneamente a la entrada de dos *buffers tri state y* las salidas de cada uno de éstos se conectan a cada una de las entradas de la MAX 232 (ver figura 4). Así, si se quiere transmitir al teléfono o al computador se habilitará dicho buffer y si se quiere enviar datos a la impresora se habilitará el otro buffer.

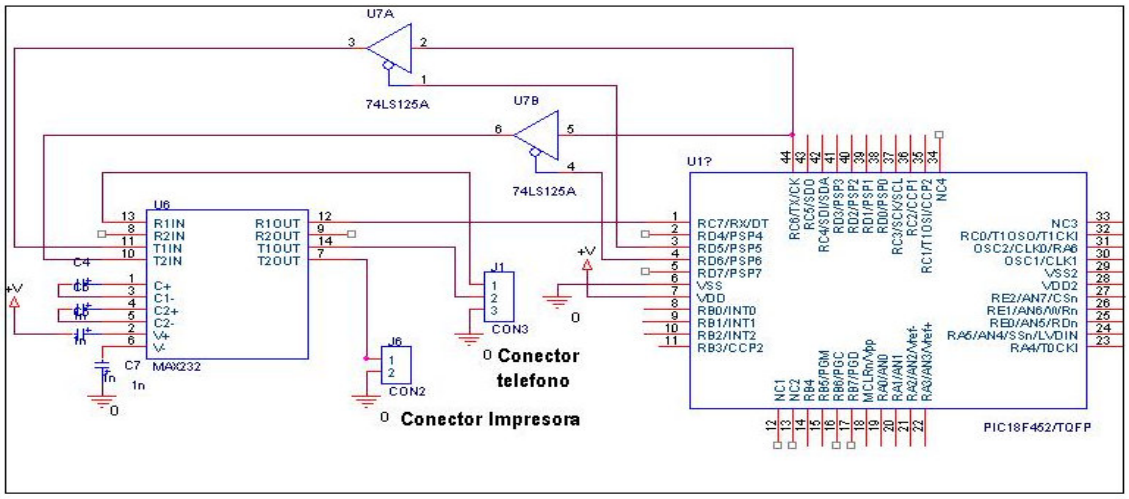


Figura 4. Conexión del puerto serial USART.

Para evitar los errores en la comunicación con el teléfono se estableció un protocolo en el que éste envía los datos al microcontrolador y el último responde un mensaje de recibido y luego ingresa a la rutina deseada. De ésta manera, desde el teléfono siempre es posible saber si la información se recibió correctamente. Para cada una de las rutinas que puede realizar el microcontrolador (Ver sección 4.1.5) existe un código específico de 8 bits de manera que éste puede ser controlado totalmente desde la aplicación desarrollada en J2ME residente en el teléfono móvil.

4.1.2 Comunicación con la impresora

El envío de datos a la impresora se hace de manera serial por el puerto de USART usando la conexión explicada en la sección anterior. Esta es una comunicación en un sólo sentido usando el protocolo RS_232 de la impresora, que cuenta con ciertos mensajes de funcionamiento (imprimir, pasar línea) y recibe los datos a imprimir en un formato ASCII.

4.1.3 Lector de banda magnética

Para el desarrollo del dispositivo se usó un lector de tarjetas de banda magnética

OMRON V3A, que es un lector de tecnología CMOS de bajo consumo con salidas compatibles con TTL. Es un lector de 9 pines que lee la PISTA I y la PISTA II (ver Anexo Marco Teórico) de las tarjetas de banda magnética. Por cada pista tiene 3 pines, uno llamado *card loaded* que se pone en un cero lógico cuando se pasa una tarjeta, el segundo es un pin de envío de datos vía serial y el tercero es el reloj de los datos. Aunque es posible realizar transacciones teniendo sólo la PISTA II de la tarjeta, se implementó la lectura de la PISTA I para que el diseño se adapte a futuros requerimientos.

Para recibir la información proveniente del lector se usaron cuatro entradas digitales (2 entradas para datos y dos de reloj) del microcontrolador. Los datos y el reloj de la tarjeta vienen negados, es decir se leen cuando el reloj esté en 0; los datos de la PISTA II vienen en bloques de a 5 bits, 4 de datos y uno de paridad (ver Anexo Marco Teórico). Para el caso de la PISTA I vienen en bloques de 7 bits (un dato de paridad). El equipo tiene una rutina para leer la tarjeta, al entrar a esta rutina se le indica cual pista se quiere leer (la 1, la 2 o ambas). De este modo el microcontrolador pasa a la rutina indicada. Se tienen tres rutinas, una para leer la PISTA I, otra para leer la PISTA II y una última que lee las dos pistas a la vez. A continuación se explican brevemente estas rutinas.

- Leer PISTA II

Al entrar en la rutina de Leer PISTA II el microcontrolador debe encontrar dentro de los datos provenientes de la tarjeta una “B” en hexadecimal que indica el comienzo de la trama. Para lograr esto, se verifica el pin del reloj hasta que se encuentre en 0, en ese momento se guarda el bit que se encuentra en la entrada de datos de la tarjeta en el bit menos significativo de un registro que previamente ha sido iniciado en 00h, luego vuelve a mirar el pin de reloj hasta que entre otro dato. Cuando el reloj se pone de nuevo en cero (es decir, cuando hay otro dato disponible), se hace un corrimiento hacia la izquierda en el registro y se pone de nuevo el dato que se encuentre en la

entrada de datos de la PISTAII en el menos significativo. Después de esto se compara el registro con el dato A0h que es el dato Bh más la paridad negados, lo que indica el comienzo de la trama de la pista.

Después de haber encontrado el comienzo de la trama se leen los datos bit por bit y se van almacenando en registros. Después de leer los cuatro bits de cada dato se revisa la paridad, en caso de ser errónea (hay un error en los datos), se envía un mensaje de error y se sale de la rutina, en caso de estar bien se compara el dato con un 0Fh que indica el final de la trama, si se leyó bien se envía por USART (hacia el teléfono) un indicador de que la tarjeta se ha leído satisfactoriamente.

- Leer PISTA I

A diferencia de la lectura de la PISTA II, la información se encuentra agrupada por bloques de 7 bits, donde el último simboliza la paridad. Para iniciar el proceso, el equipo Motorola debe enviar la instrucción para la lectura de la PISTA I. Para su lectura, la unidad de procesamiento debe encontrar un 2 en hexadecimal en el primer grupo de 7 bits y luego un 5 en hexadecimal en el siguiente grupo de 7 bits. Lo que se hace es un proceso similar a la lectura de la PISTA II, donde se hace un corrimiento y una comparación de los datos provenientes del lector de banda magnética, teniendo en cuenta el reloj para sincronizar la información. Cada vez que hay un cambio de reloj a cero lógico se toma la información de la PISTA I y se almacena, al mismo tiempo se va a haciendo un corrimiento del dato anterior. Con el nuevo dato en memoria se procede a realizar una comparación con 2 en hexadecimal; una vez se encuentra, se realiza el mismo proceso pero con el 5 en hexadecimal. Finalmente, al encontrarse los bits de inicio, se comienza a recibir la información por bloques de 7 bits, realizando la paridad con cada bloque.

Este proceso se realiza hasta que se encuentre un 0F en la información entrante, por lo que se debe estar comparando constantemente para saber si la información ya se

terminó de enviar.

Al finalizar el proceso, se envía una señal al teléfono para que continúe con la siguiente rutina.

- Leer Ambas Pistas

Para leer ambas Pistas se creó una instrucción híbrida entre la lectura de PISTA I y PISTA II. El proceso consiste en realizar el proceso de la lectura de PISTA I pero haciendo llamados al proceso de lectura de PISTA II. Se escogió como proceso principal la lectura de PISTA I debido a que ésta tiene un reloj mas rápido que la PISTA II, por lo tanto se asegura que no se pierda información de la PISTA II.

Al hacerse el proceso de llamado de la instrucción de lectura de PISTA II se controlan los estados con banderas que indican en que paso se encontraba, de ésta forma se sabe si se encuentra esperando por la B en hexadecimal o si se encuentra esperando información.

4.1.4 Teclado para ingreso de datos.

El dispositivo cuenta con un teclado numérico de 4 filas x 3 columnas que además tiene un botón de aceptado marcado A y uno para borrar marcado B. El teclado funciona como un arreglo de interruptores que ponen en corto ciertos puntos de una matriz, por tanto se uso un manejador de teclado (integrado 74HC922) cuyas ocho entradas se conectaron a las salidas del teclado y cuyas cuatro salidas se conectaron a cuatro entradas digitales del microcontrolador. Además se usó una salida de Data Available (Dato disponible) del 74HC922 que se conectó a otra entrada digital del microcontrolador y se usó una salida digital de éste último para conectarla a la entrada de Output Enable (habilitar salida) del manejador ya que éste funciona de

modo *tristate*.

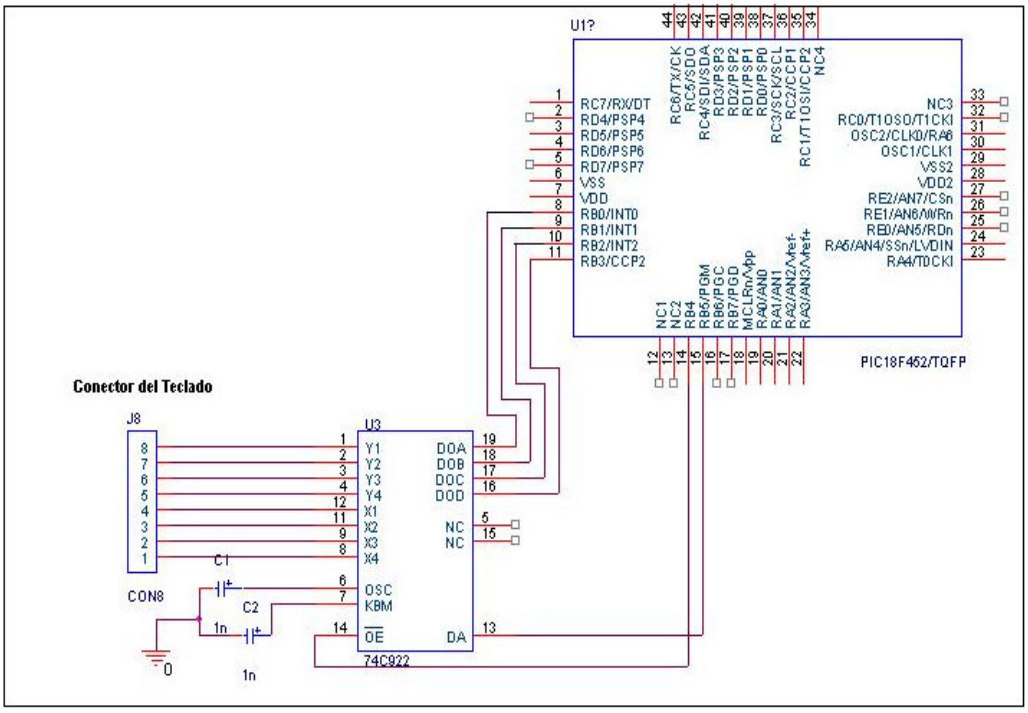


Figura 5. Conexión del teclado.

En las rutinas de recepción de datos a través del teclado se espera a que haya un 0 en Data Available, en ese momento se habilita la entrada de datos (usando el pin de output enable) se leen estos datos, se almacenan en un registro y se envían por USART (al teléfono). En caso de borrar se vacía el registro anterior y se envía un 0Bh al teléfono para que éste sepa que debe borrar. Los datos son enviados al teléfono porque se deben mostrar en pantalla para que el usuario verifique los datos que esta introduciendo. En el caso de la clave se envían asteriscos.

4.1.5 Comunicación con el modulo de cifrado.

Para comunicarse con el modulo de cifrado se utilizó el puerto serial MSSP *(Master Synchronous Serial Port*, Puerto Maestro Serial Sincrónico) del microcontrolador.

Este puerto usa tres pines SCK (*Serial Clock,* Reloj Serial) SDI (*Serial Data In,* Entrada de datos serial) y SDO (*Serial data Out,* Salida de datos serial). El puerto se programó en modo SPI (*Serial Peripheral Interface)*, de este modo se transmite en bloques de 8 bits sin bit de inicio ni de parada. La tasa de transmisión es de ¼ la frecuencia del reloj de trabajo del microcontrolador. Para el caso particular del dispositivo desarrollado es de 1,229MHZ. Además se trabajó en modo maestro, es decir el microcontrolador es quien proporciona el reloj. Las señales de entrada son muestreadas al final del tiempo de la información, la transmisión se hace con borde de subida del reloj como se muestra en la figura 6.

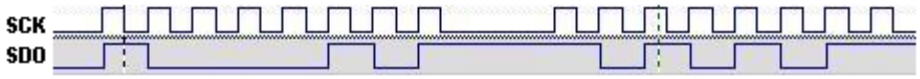


Figura 6 Transmisión de un 85h seguido de un ABh

Las señales con las que se controla el modulo de cifrado son: CLK (reloj), SERIAL CLOCK (reloj serial), D (datos), OUT DATA (salida de datos), NRESET, TAKE_D (tomar datos), DONE (realizado), ENABLE_D (datos habilitados), CRYPT (cifrar) y START. (El funcionamiento detallado del módulo de cifrado se explica en la sección 4.2). Las señales SERIAL CLOCK, OUTDATA y D corresponden en el microcontrolador a los pines del SPI. El resto de señales se implementaron en salidas y entradas digitales y se manejan de acuerdo a lo requerido. (Ver figura 7)(Ver sección 4.2).

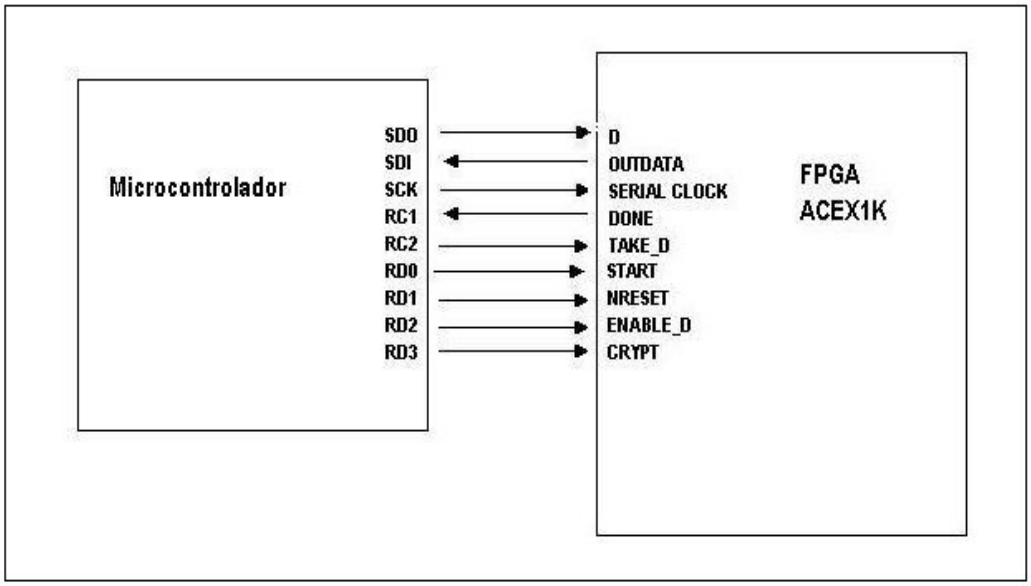


Figura 7 Conexión con el módulo de cifrado.

Se implementó una rutina en el microcontrolador para cifrar / descifrar bloques de 64 bits. En el momento de enviar un bloque de 64 bits a cifrar / descifrar el proceso que se sigue en el microcontrolador es el siguiente:

1. Se verifica con que llave se desea cifrar. (Transporte, Master, PIN). Esto se hace verificando un indicador de llave que se debe enviar como parámetro de esta rutina.

2. Se verifica si la llave esta cargada en el módulo de cifrado. El cifrador tiene la capacidad de almacenar la llave por tanto si se requiere cifrar dos bloques con la misma llave no es necesario cargar esta de nuevo. Al cargar una llave se altera una bandera dentro del microcontrolador la cual es verificada cada vez que se va a enviar a cifrar un bloque de datos.

3. En caso de no estar cargada se debe cargar la llave de datos. Se envía serialmente la llave de 64bits con las señales correspondientes (Ver sección de 4.2), éstas llaves se deben haber almacenado previamente en memoria

4. Se envía el bloque de datos de 64 bits. Se hace de manera serial por el puerto SPI.

5. Se transmite una señal de inicio y la señal de cifrado / descifrado CRYPT. (Ver sección 2.2)

6. Se verifica si se terminó el proceso de cifrado.

7. Se envía el reloj para recibir los datos de manera serial y se almacenan.

En el desarrollo se usó el modo de cifrado / descifrado CBC (ver Anexo Marco Teórico) en el cual los datos a cifrar se envían en bloques de 64 bits, pero el segundo que se transmite es la XOR del primer bloque cifrado, con el segundo bloque a cifrar y la XOR de estos con un 0C en hexadecimal, así consecutivamente. (Ver figura 8). Además tiene un vector de inicio, de manera que el primer bloque a cifrar hace XOR con este.

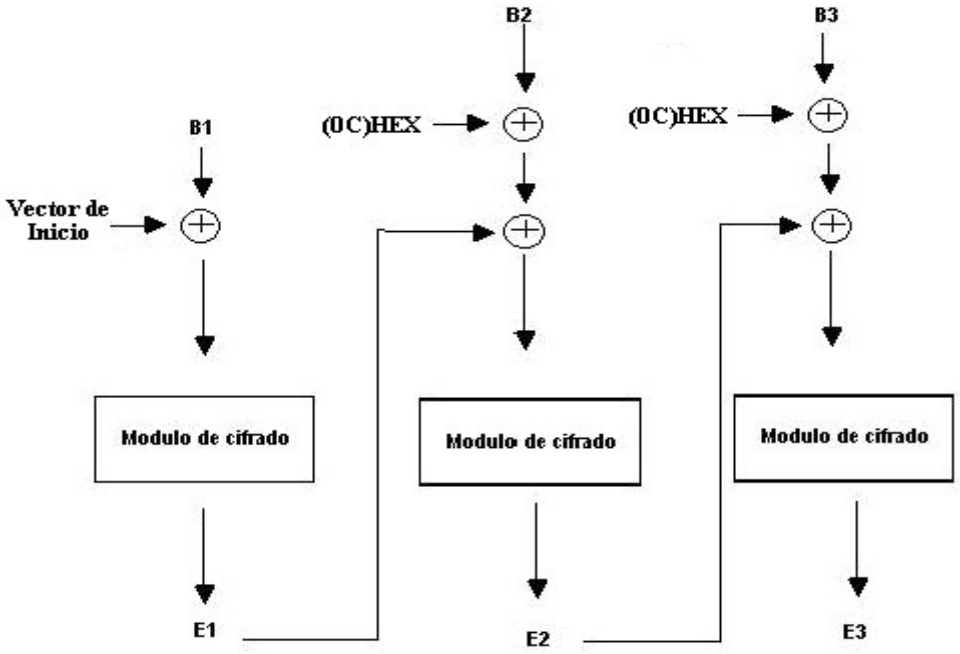


Figura 8 Cifrado usando modo CBC (Cipher Block Chaining) B1, B2, B3 son bloques de 64 bits de una misma trama, E1, E2, E3 el resultado de cifrar

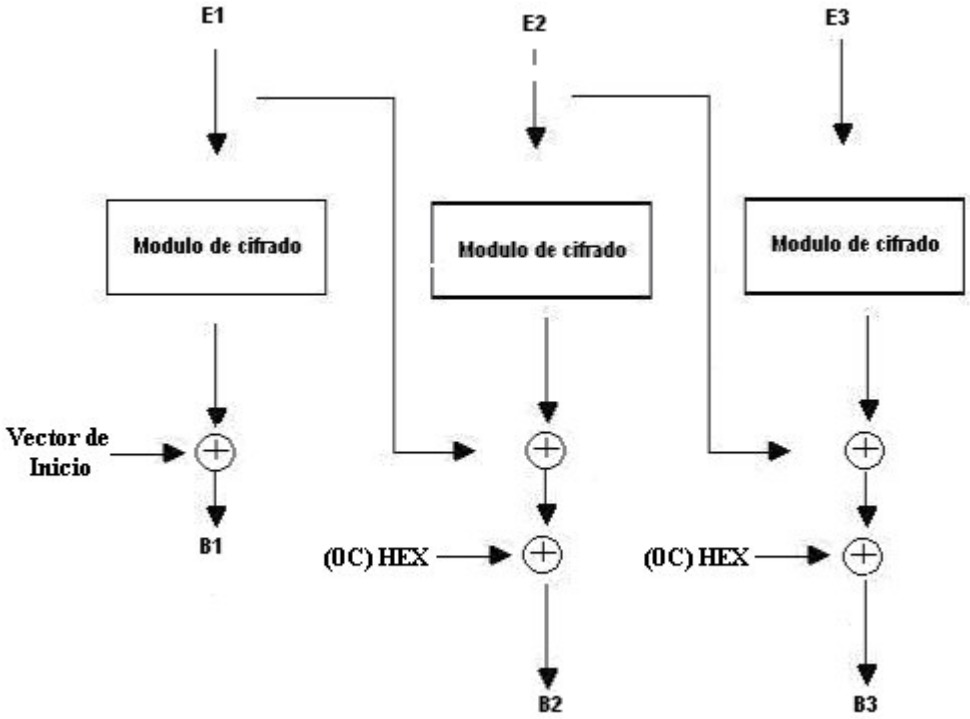


Figura 9 Descifrado usando modo CBC (Cipher Block Chaining)

De modo que este proceso de realizar la operación lógica XOR también se lleva a cabo en el microcontrolador, cada vez que se envía un bloque que no es el primero a cifrar se hace la XOR con el resultado de cifrar el bloque anterior. En el momento de descifrar se hace un procedimiento algo parecido, la diferencia consiste en que se hace la XOR del último bloque descifrado con el anterior aún cifrado.

4.1.6 Rutinas del microcontrolador.

Para hacer del dispositivo desarrollado un equipo versátil, se desarrollaron una serie de rutinas que pueden ser llamadas desde el teléfono móvil (o desde un computador en caso de ser necesario). Cada una de estas rutinas tiene un código de 8bits con la que se identifica, para llamar una rutina del microcontrolador se sigue el proceso mostrado en la figura 10. Cabe anotar que el proceso o rutina determinada puede

requerir de transmisión de información hacia el teléfono. Toda comunicación entre el teléfono (o computador) y el microcontrolador se hace a través del puerto de USART. (Ver sección 4.1.1)

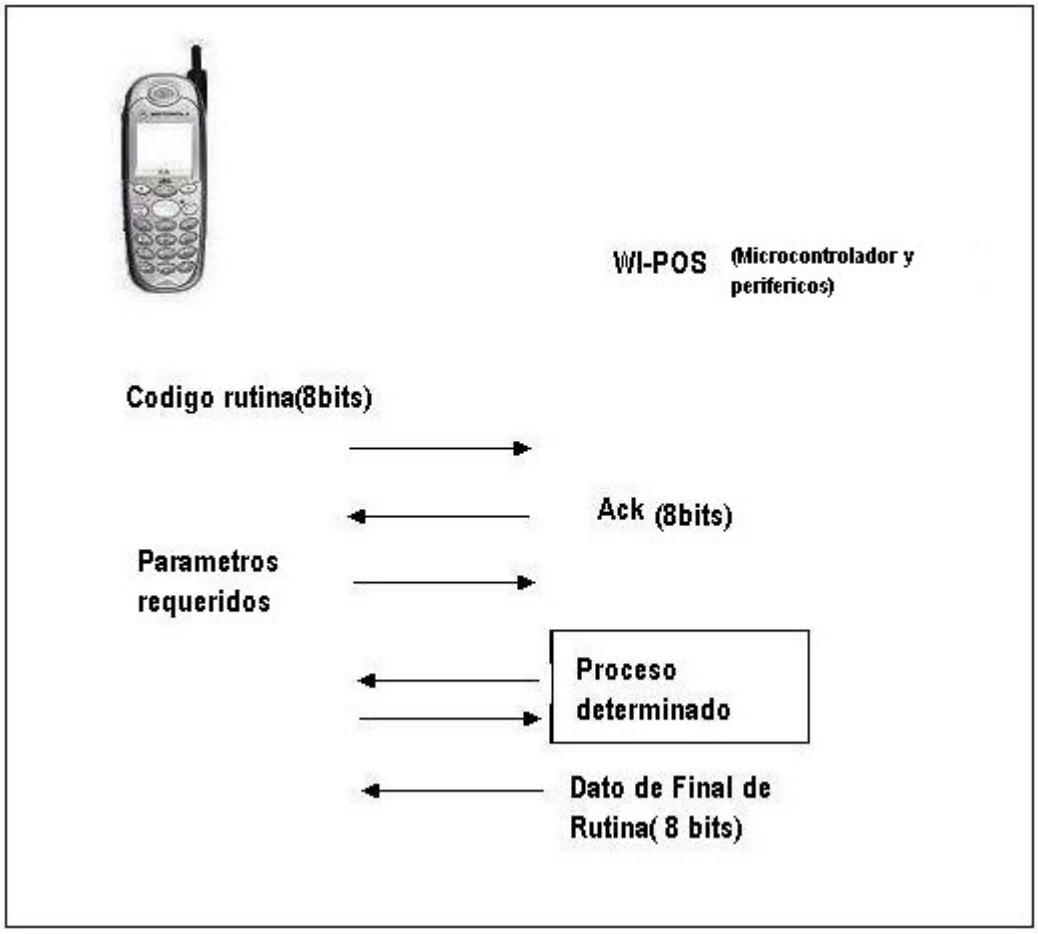


Fig. 10 Proceso para llamar las rutinas del microcontrolador.

A continuación se explicará el funcionamiento de cada rutina:

1. Leer tarjeta: En esta rutina el microcontrolador espera recibir los datos que le indican que pista leer, luego habilita los pines del lector de banda magnética, lee los datos (ver sección 4.1.3), en caso de presentarse algún inconveniente (la paridad de algún dato esta mal) devuelve un mensaje de error, de lo contrario devuelve un mensaje de aceptado y guarda los datos de la tarjeta de

banda magnética en memoria RAM.

2. Leer clave: Al recibir la instrucción de leer clave el microcontrolador espera dos parámetros que deben ser enviados desde el teléfono móvil, el primero es el numero de dígitos de la clave y el segundo el indicador de la llave con que se cifrara el PINBLOCK. Estos parámetros son necesarios porque la longitud del PIN puede variar según el país o la región o con el paso del tiempo se pueden requerir un mayor número de dígitos. Después de haber recibido estos dos parámetros el microcontrolador habilita los pines del teclado, cada vez que el usuario oprime una tecla se envía un mensaje de "asterisco" hacia el teléfono por el puerto para que éste último ponga asteriscos en la pantalla. Después de haber recibido la cantidad de dígitos necesaria por parte del usuario (dependiendo del parámetro ingresado desde el teléfono) se procede a armar una trama conocida como PINBLOCK, esta es una trama que se construye haciendo operaciones lógicas con el PIN del usuario y el número de la cuenta, por esto es necesario que para correr esta rutina ya se haya deslizado la tarjeta. El resultado (una trama de 64 bits) se cifra usando la llave cuyo indicador que se ingresó como parámetro y se almacena en memoria RAM. El PINBLOCK se cifra inmediatamente porque es un dato que requiere de total confidencialidad, para evitar problemas de seguridad se almacena debidamente cifrado.

3. Obtener datos: Esta es una rutina en la que se almacenan datos provenientes del teclado. Sirve para almacenar datos tales como valor, cuotas y propina. El microcontrolador cuenta con 10 posiciones de memoria para almacenar datos, cada una de 10 bytes. Al entrar en esta rutina el microcontrolador espera dos parámetros, el primero es el número máximo de dígitos del dato a recibir y el segundo el índice en donde se guardará el dato (éste último se encuentra entre 1 y 10). Luego se habilita el teclado, se leen los datos, se envían al teléfono cada vez que se oprima una tecla para que puedan ser mostrados en pantalla. Este proceso se realiza hasta que se haya ingresado el número máximo de

dígitos o el usuario presione la tecla “A” del teclado. Los datos son almacenados en memoria RAM y la posición donde se almacenan depende del índice introducido como parámetro.

4. Pedir tarjeta: Esta rutina envía por USART los datos de la PISTA 2 de la tarjeta que debe haber sido deslizada previamente.

5. Pedir datos: Esta rutina recibe como parámetro el índice de los datos que se requiere (entre 1 y 10) y devuelve los datos que fueron almacenados haciendo uso de la rutina obtener datos.

6. Cifrar: Esta rutina espera dos parámetros, primero espera el bloque de 64 bits a cifrar y segundo el indicador de la llave a utilizar. Luego cifra el bloque de 64 bits haciendo uso del modulo de cifrado y almacena los datos en memoria RAM. Cabe anotar que como las llaves se almacenan debidamente cifradas (usando la llave maestra) en el proceso de cifrado se debe descifrar primero la llave a usar para luego enviarla al módulo de cifrado. El cifrado/descifrado de las llaves se hace de la misma manera que con cualquier bloque de 64bits.

7. Descifrar: Esta rutina funciona de manera similar a la de cifrar, la diferencia consiste en que se elige otro modo de operación del módulo de cifrado haciendo uso del pin CRYPT (Ver Sección 2.2).

8. Reset: Esta instrucción inicializa el microcontrolador.

9. Recibir Terminal: Esta rutina se debe correr desde un computador personal y no desde el teléfono. El número de terminal es un código que identifica cada equipo y que es necesario para llevar a cabo una transacción. En esta rutina se recibe el número de terminal desde el computador (8 bytes) y se almacena en la memoria EEPROM del microcontrolador. Este es un código que se utiliza en todas las transacciones y por eso es necesario cargarlo en una memoria no

volátil.

10. Recibir llave de transporte: Esta rutina se corre desde un computador y no desde el teléfono móvil. En esta rutina se recibe de manera serial la llave de transporte, se cifra usando la llave maestra y se almacena en memoria EEPROM del microcontrolador, esta llave se usa para el cifrado de las tramas que se envían al servidor.

11. Recibir llave maestra: La recepción y almacenamiento de la llave maestra requiere de varias rutinas, este proceso se explica detalladamente en la sección 4.4.

12. Parámetros iniciales: Para correr esta rutina de manera correcta se debe haber cargado el número de terminal, la llave maestra y la llave de transporte. Al entrar en esta rutina el microcontrolador construye la trama de petición de parámetros iniciales (ver sección 3.3.1), rellena con el carácter "F" los bytes restantes necesarios para tener una trama que sea múltiplo entero de 64 bits (para que pueda ser cifrada). Luego cifra esta trama haciendo uso de la llave de transporte usando el modo de cifrado CBC y la envía al teléfono, después queda en estado de espera de los parámetros iniciales. Al recibir la trama de parámetros iniciales la descifra usando la llave de transporte, verifica que no haya errores en el mensaje, esto lo hace verificando ciertos campos del mensaje recibido (tipo de mensaje, terminal) en caso de no haber errores almacena los parámetros en memoria RAM y envía un mensaje de aceptado. En caso de errores se envía un mensaje de error. Esta rutina se debe realizar cada vez que se enciende el equipo porque los parámetros iniciales son indispensables para cualquier transacción.

13. Pedir Parámetros iniciales: Es una serie de rutinas que envían al teléfono cualquiera de los parámetros iniciales. Existe una rutina para cada uno de los parámetros, al entrar en ésta el microcontrolador envía la información

almacenada en la posición de memoria de ese parámetro. (Para ver cuales son los parámetros iniciales ver la sección 3.3.2)

14. Parámetros de tarjeta: Para correr esta rutina se deben haber cargado previamente los parámetros iniciales y además se debe haber deslizado la tarjeta de banda magnética. En este proceso el microcontrolador construye la trama denominada "petición de parámetros de tarjeta", rellena con el carácter "F" hasta tener una trama múltiplo entero de 64 bits, la cifra usando el modo CBC y la envía al teléfono, luego queda en estado de espera de respuesta. Al recibir la trama de respuesta de parámetros iniciales se descifra, verifica que este correcto el tipo de mensaje y almacena los parámetros de la tarjeta en ciertas posiciones de memoria RAM establecidas para esto.

15. Pedir parámetros de tarjeta: Similar a la petición de parámetros iniciales se implementó una serie de rutinas para pedir cada uno de los parámetros de la tarjeta, en las que el microcontrolador envía a través del puerto la información almacenada en las posiciones de memoria destinadas para los parámetros de la tarjeta.

16. Petición de llave de cifrado de PIN: Esta rutina se encarga de construir la trama de petición de llave de cifrado PIN; la cifra usando la llave de transporte y la envía, luego queda en estado de espera. Cuando recibe la respuesta descifra la trama, verifica el tipo de mensaje y el terminal, cifra la llave de cifrado de pin usando la llave maestra y la almacena en memoria RAM. Como la llave de cifrado de PIN se pide durante cada transacción no es necesario almacenarla en EEPROM.

17. Solicitud de transacción de pago: Para ejecutar correctamente esta rutina se deben haber solicitado parámetros iniciales, parámetros de tarjeta y todos los campos adicionales dependiendo de la tarjeta (valor, cuotas, PIN, tipo de cuenta entre otros, ver sección 3.3). Al tener estos datos, el microcontrolador

construye la trama de petición de transacción de pago y la envía por el puerto serial, luego queda en estado de espera. Al recibir la respuesta la descifra, revisa errores (en caso de error transmite un mensaje específico); de estar correcto, almacena los datos de respuesta de transacción de pago en los campos de memoria RAM destinados para ello.

18. Petición de respuesta de transacción: Se implementó una serie de rutinas para pedir cada uno de los campos de la respuesta de transacción de pago, en las que el microcontrolador envía a través del puerto la información almacenada en las posiciones de memoria destinadas para los parámetros de la tarjeta.

19. Transacción de reverso: En esta rutina se construye la trama de transacción de reverso, se cifra y se envía, luego se espera recibir la respuesta, se verifica que no tenga errores y almacenan los datos en memoria.

20. Inicializar consecutivo: El consecutivo es un número que caracteriza cada una de las transacciones. Esa rutina pone en cero los seis registros en EEPROM correspondientes al consecutivo.

21. Aumentar consecutivo: Esta rutina le suma 1 a los registros del consecutivo, teniendo en cuenta que los datos son de tipo numérico, es decir van entre 0 y 9 (ver sección 3.3.5).

22. Comparar fecha de vencimiento: Esta rutina compara una fecha ingresada desde el teléfono con la fecha de vencimiento que se encuentra en la PISTA II de la tarjeta. La fecha de vencimiento de la tarjeta debe ser mayor a la fecha ingresada por el usuario, de no ser asi se envia un mensaje de error.

23. Comparar fecha de vencimiento 2: Esta rutina compara una fecha ingresada por el usuario con la fecha de vencimiento de la tarjeta que se encuentra en la Pista II de la tarjeta. Envía un mensaje de respuesta, dependiendo del

resultado, en este caso las fechas deben ser iguales .

4.2. Cifrado de datos

Una parte fundamental del trabajo de grado y sin la cual un dispositivo como WI-P.O.S no tendría validez alguna en el mercado es el módulo de cifrado, ya que es la parte que se encarga de darle un nivel de seguridad a la información cuando viaja por la red.

Debido a que WI-P.O.S debe ser un dispositivo que cumple con los estándares de seguridad, su módulo de cifrado se basa en un algoritmo DES (*Data Encryption Standard*), este sistema de cifrado es utilizado en el mercado colombiano por la complejidad que implica el tratar de descifrarlo sin tener el acceso a las llaves.(ver Anexo Marco Teórico)

Para su implementación se evaluó que la mejor forma de hacerlo era mediante el uso de una FPGA (*Field Programmable Gate Array*, Arreglo de compuertas programables), debido a que por medio de este dispositivo se tenía la posibilidad de manipular directamente los 64 bits de la llave y de la información.

La FPGA seleccionada fue una ACEX 1K50TI144-2, esta consta de 50 mil compuertas lógicas, teniendo aproximadamente 5000 elementos lógicos programables, tiene un empaque de montaje superficial delgado y un total de 144 pines. Una de las características de las FPGA es que tienen memoria volátil, por lo que es preciso utilizar una memoria para recargar los parámetros del programa desarrollado, en nuestro caso el cifrado en DES, de lo contrario, cada vez que se prendiera la FPGA tendría que programarse directamente de la herramienta de desarrollo. Por esto se implementaron las conexiones correspondientes entre la ACEX 1K y una memoria serial de ALTERA EPC2LC20. A continuación se hace una descripción de las conexiones en la Figura 11, el tipo de conexión utilizado es llamado *Passive Serial*, o Serial Pasivo.

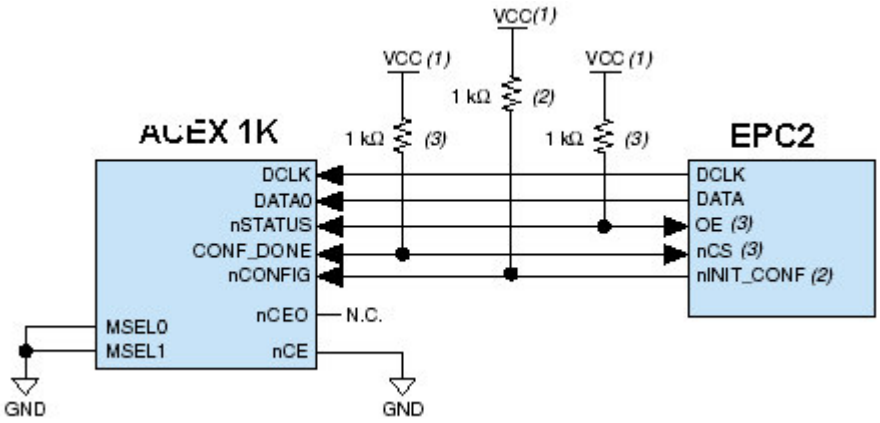


FIGURA 11, Configuración Serial Pasiva entre la memoria EPC2 y la FPGA ACEX 1K[3]

A parte de las conexiones entre memoria y FPGA se deben crear conexiones al programador, que en este caso es JTAG (*Joint Test Action Group*), que es usado por las herramientas de desarrollo de ALTERA. A continuación se muestran las conexiones correspondientes en la Figura 12 y en la Tabla 1 .

[3]Para un mayor entendimiento favor referirse a la documentación de la ACEX1K y la memoria EPC2 en la página www.altera.com

Pin	PS Mode	JTAG Mode
	Signal	Signal
1	DCLK	TCK
2	GND	GND
3	CONF_ DONE	TDO
4	VCC	VCC
5	nCONFIG	TMS
6	–	–
7	nSTATUS	–
8	–	–
9	DATA0	TDI
10	GND	GND

TABLA 1 Conexiones ente ACEX1K y JTAG en modo Serial Pasivo[4]

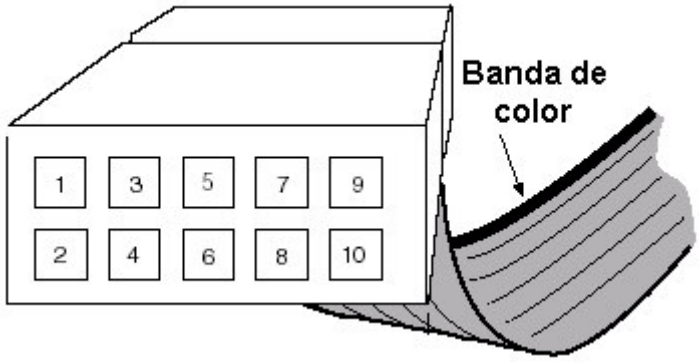


Figura 12, conector JTAG para programar la ACEX1K

Con la información de la figura y la tabla anterior se pueden establecer las conexiones, cabe anotar que los componentes electrónicos con que se trabajó se encuentran disponibles solo en montaje superficial, lo que hace que sus pruebas se

[4] Para manyor información acerca de las conexiones favor remitirse a la pagina www.altera.com

deban realizar directamente en el impreso, por esto se debe tener un especial cuidado en las conexiones y en el diseño del impreso.

Para el desarrollo de DES se siguió el proceso establecido según el documento FIPS 46, explicado en el marco teórico, se evaluaron las posibles herramientas para su implementación como MAX PLUS II, MODELSIM y QUARTUS II, y finalmente se decidió implementar en esta última, por la facilidad y comprensión de la herramienta y por que las FPGAs que tenía cumplían con las características apropiadas para la implementación del módulo de DES.

Debido a que el dispositivo se tiene que comunicar con el PIC por medio de su puerto SPI, la información de los datos y de la llave se deben cargar serialmente, de manera que se implementaron 2 módulos aparte del de DES, estos son: un módulo que convierte la información de serie a paralelo para poder recibir la información en el módulo DES y otro módulo que convierte la información de paralelo a serie para poder enviar la información ya cifrada. Cabe anotar que estos módulos fueron implementados dentro de la misma FPGA. En la figura 13, se puede observar la conexión de los módulos entre ellos.

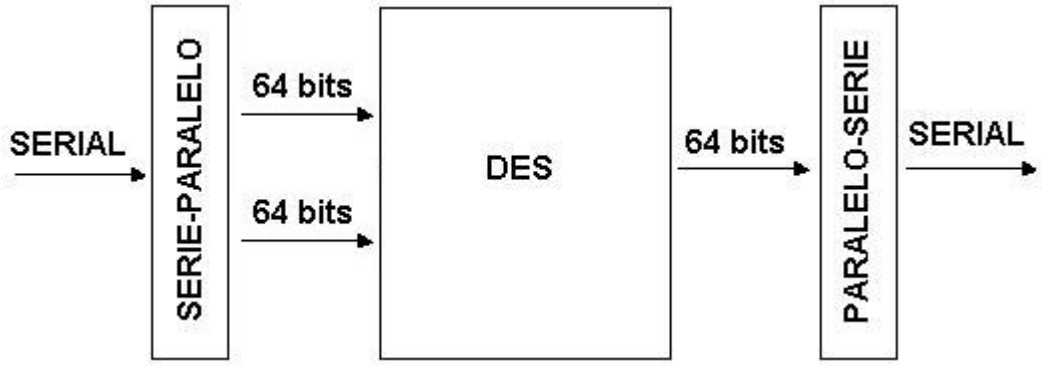


FIGURA 13, Módulos de entrada y saluda a DES

4.2.1. **Módulo Serie-Paralelo**

Este módulo recibe la información que proviene de forma serial y la convierte en paralelo, debe entender en que momento se envía la información de la llave o de los

datos a cifrar, también debe ser enviado con su respectivo reloj para poder interpretar la información serial.

Es por esto que se han definido 3 señales de entrada importantes en este módulo, los cuales son:

- D: Por esta entrada se envían los datos.
- SERIAL_CLOCK: Por esta entrada se envía el reloj para que el módulo interprete los datos.
- NRESET: Con esta señal se puede dar reset al módulo, borrando sus registros y sus contadores internos. Cabe anotar que esta señal es negada.
- ENABLE_D: Este Pin identifica si la información enviada pertenece a la llave o a los datos a ser cifrados.
  1: Datos a ser cifrados.
  0: Llave.
- KEY (0..63): Salida donde se almacena la llave.
- KEY READY: Salida que indica que la llave fue cargada.
  1: Llave cargada
  0: Llave no cargada.
- DATA (0..63): Salida donde se almacenan los datos a ser cifrados.

En la Figura 14 se pueden ver las entradas y las salidas del módulo.

FIGURA 14, Módulo serie-paralelo

Para enviar la información serial se deben seguir las siguientes especificaciones: primero se debe enviar un 01h para poder comenzar la transmisión de datos al módulo Serie-Paralelo, todo sincronizadamente con la señal de reloj. Desde este punto el módulo cuenta 64 ciclos de reloj para saber que la información ha llegado completa. Si la información que se envió pertenecía a los datos, la entrada de ENABLE_D debió haber permanecido en 0 durante el envío de la información. En el momento de ser la llave cargada, un contador interno cuenta los 64 ciclos de reloj de la carga de la llave y en el momento de terminar de contar levanta la señal de KEY READY. Si la información que viajaba eran los datos a ser cifrados ENABLE_D debió haber permanecido en 1 durante el envío de la información. En la Figura 15, se observa cuando el módulo recibe la información de los datos a cifrar y en la Figura 16 se puede ver cuando recibe la información de la llave.

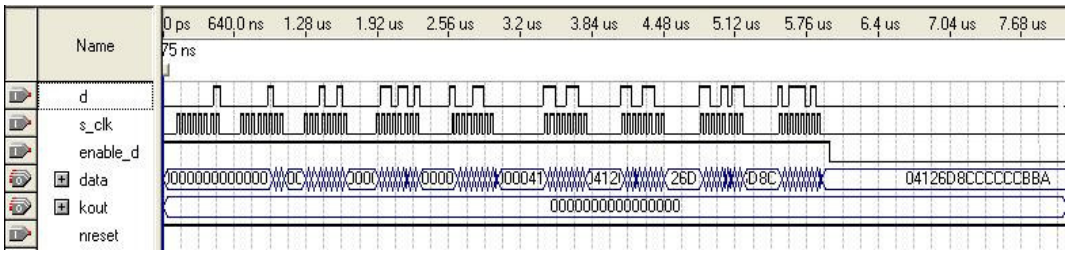


FIGURA 15, Simulación de módulo Serie-Paralelo con recepción de datos a cifrar

En el ejemplo de la Figura 16, se puede observar que para enviar la información se necesita primero enviar el 01h para que el sistema se prepare a recibir los siguientes 8 Bytes, también la señal de ENABLE_D se encuentra en todo momento en 1 al igual que la señal de NRESET. Al finalizar el envío de la información se puede ver en la salida data la información en paralelo.

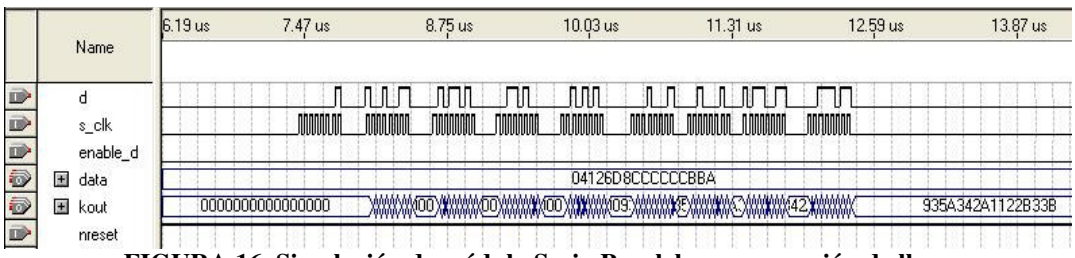


FIGURA 16, Simulación de módulo Serie-Paralelo con recepción de llave

En la Figura 16, se muestra el envío de la información de la llave, por lo que, en lo único que difiere del ejemplo de la Figura 15, es en la señal ENABLE_D que se encuentra en 0.

4.2.2. Módulo Paralelo-Serie

Este módulo se encarga de recibir la información de 64 bits en paralelo ya cifrada del módulo DES y volverla serial para poderla enviar al microcontrolador. Recordando que todo el módulo de cifrado funciona como esclavo del microcontrolador la información que entrega el módulo paralelo-serie, depende del reloj enviado por el maestro.

Sus entradas y salidas son las siguientes:

- Q (63..0): En esta entrada se pone la información paralela que será convertida en serie por el módulo, es decir, el resultado de la realización del cifrado en el módulo de DES.
- TAKE_D: Con esta entrada se le indica al sistema cuando tomar los datos que se encuentran en su entrada Q (63..0).
- SERIAL_CLOCK: Esta entrada pertenece al reloj generado por el maestro quien controla en que momento se le debe enviar la información serial.
- NRESET: Con esta señal se puede dar RESET al módulo, borrando sus registros. Cabe anotar que esta señal es negada.
- OUTDATA: En esta salida la información ingresada paralelamente al sistema sale de forma serial, sincronizada con la señal de SERIAL_CLOCK enviada por el maestro, cabe anotar que la información es enviada con el borde de subida de la señal de reloj.

En la Figura 17, se pueden ver las señales del módulo paralelo-serie.

FIGURA 17, Módulo paralelo-serie

Para solicitar al módulo la información se debe primero poner en 1 por 8 ciclos de reloj la señal TAKE_D, esto, debido a que con esta señal se cargan los registros de entrada almacenando la información que se encuentra en Q (63..0), luego se solicitan los datos cifrados con sólo enviar el reloj. A continuación en la Figura 18, se muestra un ejemplo de cómo el módulo paralelo-serie entrega la información cifrada.

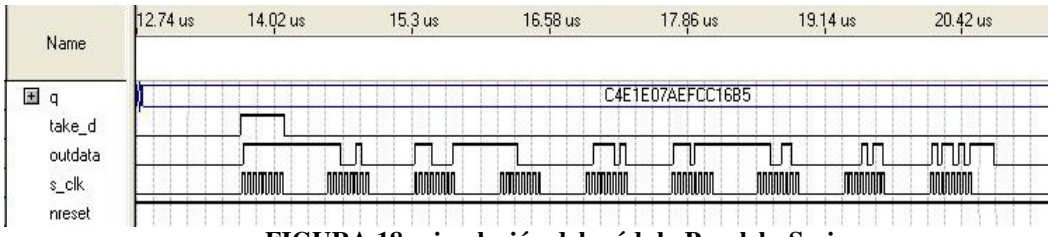


FIGURA 18, simulación del módulo Paralelo-Serie

4.2.3. Modulo DES:

Para la elaboración de este módulo se siguieron los pasos establecidos en el Marco Teórico en la sección de DES, donde se explica el funcionamiento de este sistema de cifrado.

Para su implementación se desarrollaron diferentes funciones en el lenguaje de ALTERA AHDL, en donde se repartieron los desarrollos de las tablas de permutación, sustitución y otros, para distribuir el módulo DES de una manera más sencilla de desarrollar.

4.2.4. Funciones de Tablas:

A continuación se explican las funciones de las tablas desarrolladas para la implementación de DES, para un mayor entendimiento del funcionamiento de cada tabla se recomienda referirse al marco teórico.

- IP (Initial Permutation)
- IP^{-1} (Inverse Initial Permutation)
- PC1 (Permuted Choice 1)
- PC2 (Permuted Choice 2)
- E (Expantion)
- P (Permutation)
- S_n (Tablas de Sustitución)

- **IP (*Initial Permutation*):**

La permutación inicial es realizada sobre la información de 64 bits a ser cifrada, como salida se tienen 2 bloques de 32 bits cada uno denominados L0 y R0, este proceso se puede ver claramente en el marco teórico. En su implementación en AHDL la entrada se reordena según la tabla IP, utilizando para esto funciones de tablas por su facilidad de manejo.

- **IP^{-1} (*Inverse Initial Permutation*):**

La permutación inicial inversa es hecha sobre el último bloque $R_{16}L_{16}$ teniendo como entrada un bloque de 64 bits, estos bits son reordenados según la tabla **IP^{-1}** del

- **E (*Expantion*):**

La tabla de Expansión se aplica sobre el bloque R_n en cada iteración, tiene una

entrada de 32 bits y una salida de 48 bits, de ahí el nombre que recibe.

- **P (*Permutation*):**

La tabla de permutación P se aplica sobre la información que sale de las tablas de sustitución, tiene como entrada 32 bits y como salida 32 bits.

- **PC-1(*Permutation Choice 1*):**

Esta tabla tiene como entrada el bloque de 56 bits de la llave sin sus bits de paridad y como salida tiene 2 bloques de 28 bits cada uno, denominados en el Marco teórico como C0 y D0.

- **PC-2(*Permutation Choice 2*):**

Esta tabla tiene como entrada la unión de los bloques C_n y D_n después de su corrimiento, tiene como entrada un bloque de 56 bits y como salida un bloque de 48 bits.

- **Tablas de sustitución**

Estas tablas tienen como entrada un bloque de 6 bits, donde el primer y último bit significan el número de la fila de la tabla, que va de 0 a 3 en binario. Los bits de la mitad simbolizan el número de la columna y pueden tomar los valores de 0 a 15 en binario.

4.2.5. Funciones Específicas:

A partir de la definición e implementación de cada una de las tablas anteriores por

medio de funciones se generaron otras funciones más específicas para generalizar los pasos.

- **Función de sustitución:**

Al tener implementadas las funciones de las tablas de sustitución se creó una función más general en la que se invocaban todas, teniendo como entrada un bloque de 48 bits y una salida de 32 bits. La función divide el bloque de 48 bits de entrada en pequeños grupos de 6 bits, en donde el primer grupo entra a la primera tabla de sustitución, el segundo a la segunda tabla y así sucesivamente. Como cada tabla de sustitución genera como respuesta bloques de 4 bits, éstos son unidos al final, generando un total de 32 bits al agruparlos.

- **Función F:**

Esta función, definida en el marco teórico, tiene como entradas dos bloques, uno de 32 bits que representa R_n y otro de 48 bits que representa la subllave, como salida tiene un bloque de 32 bits. Esta función introduce el bloque R_n en la función de la tabla de Expansión E, generando una salida de 48 bits, que son operados con la subllave por una XOR. Estos 48 bits resultantes son introducidos a la función de sustitución explicada anteriormente, teniendo como salida un bloque de 32 bits. En este punto se ve la importancia de haber desarrollado el sistema de cifrado por funciones.

- **DES:**

Para la generación del algoritmo DES se utilizaron los recursos de memoria de la FPGA para almacenar las subllaves generadas para cada función F, incluyendo el archivo de manejo de memorias llamado “lpm_ram_dq.inc”, donde se almacenaron 16 llaves cada una de 48 bits. Esta librería permite manejar memorias con tamaño de

información variable, para el caso de la generación de las subllaves fue conveniente porque se utilizaron 48 bits de tamaño de dato.

Para la generación de las subllaves, el bloque de 64 bits de entrada fue primero pasado por la función PC-1, la cual entregaba 2 bloques de salida C_0 y D_0, luego como la generación dependía de corrimientos a la izquierda, los 2 bloques fueron introducidos en una máquina de estados que realizaba los corrimientos según la iteración en que se encontraba. Al realizarse los bloques Cn y Dn eran unidos y se llamaba la función PC-2 donde finalmente se tenía la subllave generada. Con esto se almacenaba la información en memoria.

Para cifrar la información, se cogían los datos y se llamaba la función IP, con este resultado se generaron los bloques L_0 y R_0 los cuales eran pasados por la función F creada anteriormente. Para las iteraciones se creó una máquina de estados donde en cada iteración se hacían los cambios de los bloques L y R y además se llamaba la función F invocando la subllave de la memoria.

4.2.6. Entradas del módulo

- KEY (64..1): Entrada paralela de la llave de 64 bits.
- DATA (64..1): Entrada paralela de la información a cifrar o descifrar.
- CRYPT: Señal que indica si la operación que se va a realizar es de cifrado o descifrado.

  1: Cifrado.

  0: Descifrado.
- START_KEY: Entrada, esta señal le indica al módulo DES que comience a generar las subllaves.
- START: Entrada esta señal le indica al módulo DES que comience a cifrar la

información.

- CLOCK: Reloj del módulo.
- NRESET: Señal negada que le da reset al sistema

  1: No reset.

  0: Reset.
- OUT_DES (64..1): Salida paralela de la información ya cifrada o descifrada.

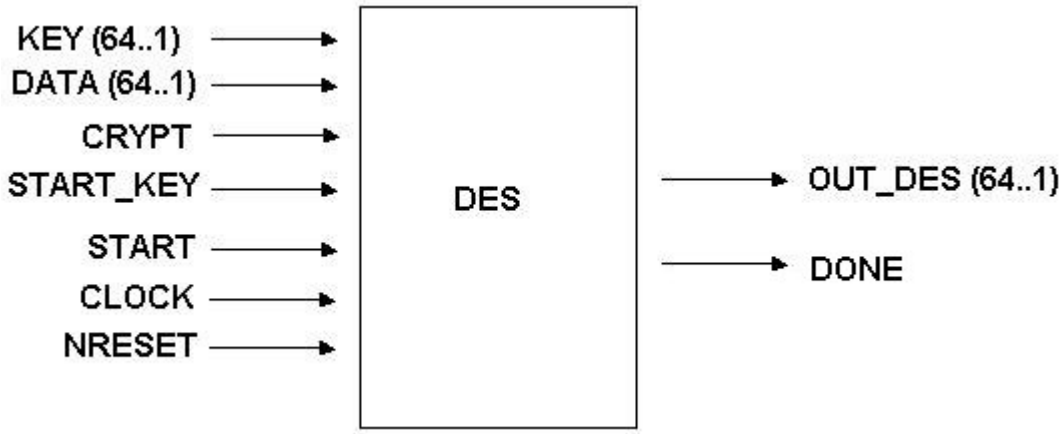


FIGURA 19, Entradas y Salidas del modulo DES

4.2.7. Integración de los módulos:

De acuerdo a la descripción de los módulos explicados anteriormente, son unidos por medio de sus señales entre sí. El módulo serie-paralelo se interconecta con el módulo DES por medio de las señales de la salida paralela de la llave de 64 bits y la salida de la información a cifrar de 64 bits. La señal de KEY_READY se encarga de dar la orden al módulo DES de comenzar a generar las subllaves, ésta señal se conecta con START_KEY.

El módulo DES se conecta con el módulo paralelo-serie por medio de la salida paralela de OUT_DES. Las conexiones entre cada módulo se pueden ver en la Figura19.

4.2.8. Módulo de Cifrado

Teniendo los módulos interconectados se genera el módulo de cifrado, el cual tiene las siguientes señales de control.

- NRESET: Esta señal de entrada se encarga de reiniciar los módulos y borrar la información almacenada en sus registros.
  1: No reset.
  0: Reset
- ENABLE_D: Esta señal de entrada indica que información se va a enviar al módulo por la entrada D.
  0: La información que entra a D pertenece a la llave.
  1: La información que entra a D pertenece a la información a cifrar.
- D: Esta entrada recibe la información serialmente, ya sea la llave o la información a cifrar.
- SERIAL_CLOCK: Reloj que sincroniza la información que entra por la entrada serial D.
- START: Con esta entrada se da la orden al módulo de comenzar a cifrar.
- CLOCK: Reloj del módulo de cifrado.
- CRYPT: Señal de entrada que indica si se va a realizar una operación de cifrado o descifrado.
  1: Cifrado.
  0: Descifrado.
- TAKE_D: Esta señal de entrada prepara al módulo de cifrado para enviar la información del resultado del cifrado o descifrado.
  1: Enviar información.
  0: No enviar la información.
- OUT_DATA: Señal de salida por donde se envía la información.
- DONE: Señal de salida que indica que la información ya se terminó de cifrar o descifrar

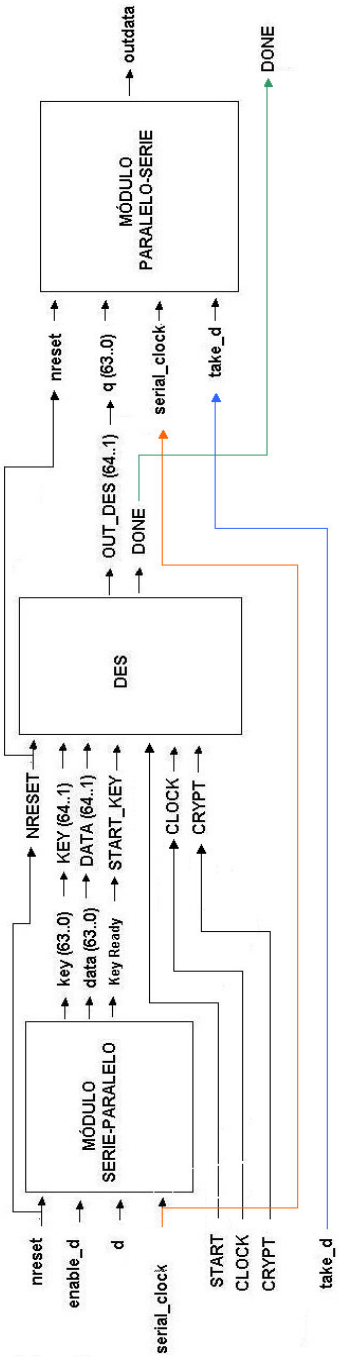


FIGURA 20, Interconexión entre módulos

4.3 APLICACIÓN EN J2ME PARA EL TELÉFONO MÓVIL MOTOROLA

Para el uso apropiado del dispositivo es necesario tener una interfaz gráfica amable que es la encargada de interactuar con el usuario para llevar a cabo la transacción con

éxito, ya que esta interfaz le permite al usuario llevar el control de la transacción paso por paso ingresando la información necesaria de la forma como la aplicación se la va solicitando.

Esta aplicación fue desarrollada en el lenguaje de programación *Java 2 Micro Edition* (J2ME), que está orientado a dispositivos móviles con limitada capacidad de procesamiento y de memoria como los teléfonos móviles y PDA's, y que está siendo incorporada de fábrica en los equipos móviles Motorola iDEN de Avantel a los cuales está orientado WI-P.O.S.

4.3.1. Descripción de la aplicación:

Existen dos aplicaciones importantes que son las que permiten llevar a cabo con éxito una transacción bancaria utilizando el dispositivo desarrollado en este Trabajo de Grado. Hay una tercera aplicación que consiste en generar el reverso de una transacción, es decir anula la ultima transacción realizada, esto por si existe algún tipo de error.

La primera de ellas es la encargada de permitir cargar los Parámetros Iniciales(ver sección 3.3) al dispositivo. Es una aplicación en la cual no interactúa el usuario que va a realizar un pago a través de tarjeta débito o tarjeta de crédito sino que la ejecuta el funcionario del establecimiento cada vez que se enciende el equipo.

La segunda aplicación es la aplicación de Transacción de Pago, la cual permite realizar un pago con tarjeta débito o tarjeta de crédito.

Las aplicaciones mantienen un contacto permanente con el dispositivo WI-P.O.S por medio de un puerto serial que tiene el teléfono y también se comunican con el servidor que recibe y envía información acerca de la transacción a través del protocolo seguro de Internet HTTP (*HyperText Transfer Protocol*), utilizando la red iDEN de Avantel. Las aplicaciones están divididas en clases, propias del lenguaje

Java que es orientado a objetos, las cuales permiten un manejo completamente modular ya que hay un programa principal donde se desarrolla la secuencia de cada aplicación y en ese programa principal se hacen diferentes llamados a estas clases para que realicen acciones específicas, y a veces repetitivas, reduciendo la cantidad de código para la aplicación.

4.3.2 Clases desarrolladas en la aplicación:

- *CommPuerto:* Permite la comunicación del dispositivo móvil con el puerto serial para enviar y recibir información, manejando número enteros. Esta clase recibe tres parámetros de tipo entero (int). El primero es el dato a enviar por el puerto serial, el segundo es un entero que indica cuantos bytes espera recibir la aplicación de la unidad central de procesamiento y el tercero es un entero que indica si se envía un dato o solamente se espera una respuesta. Cada byte que esta clase recibe del Wi – POS, lo convierte a un número entero. Al final, devuelve una cadena de caracteres, con números enteros, a la aplicación principal.

- *CommPuerto2:* Permite la comunicación del dispositivo móvil con el puerto serial para enviar y recibir información, manejando información en hexadecimal. Recibe los mismos tres parámetros que CommPuerto pero la diferencia es que cada byte que recibe, lo expresa en una cadena hexadecimal. Al final devuelve una cadena de caracteres, con números en hexadecimal, a la aplicación principal.

- *DataRequest:* Permite el envío de instrucciones a la unidad central de procesamiento para que guarde información ingresada por el teclado y esta información se vea en la pantalla del teléfono móvil. Esta clase recibe como parámetros un entero que expresa cuantos dígitos se esperan recibir del teclado, un objeto de tipo “Forma” que es la pantalla que se muestra en ese momento al usuario en la cual van a ir apareciendo los números que se digiten

por el teclado del Wi – POS, y un índice que le indica a la unidad central de procesamiento en qué lugar de memoria se debe guardar esa información. Al final, devuelve una cadena de caracteres indicando si la operación fue exitosa o no.

- *Flag, positionThread:* Estas clases permiten el manejo de "pantallas temporales", muy usadas en la aplicación, para mostrar al usuario una información mientras el equipo realiza otra tarea. Una vez finalizada esa otra tarea, se cambia la pantalla a una pantalla final. Un ejemplo de esto es cuando la aplicación muestra la pantalla que le indica al usuario que deslice la tarjeta por el lector. Mientras que el Wi – POS espera que esto ocurra, el teléfono muestra esa pantalla y apenas el usuario desliza la tarjeta, el Wi – POS le indica a la aplicación que la tarjeta fue deslizada y la aplicación muestra la pantalla siguiente. Son pantallas que permiten al usuario ver que el sistema se encuentra realizando algún proceso, cuando éste puede tardar varios segundos. Si ésta clase de pantallas no se incorporaran en la aplicación, el usuario podría confundir una pantalla congelada temporalmente con un bloqueo de la aplicación.

- *Httpconnection:* Permite la conexión a Internet, enviando y recibiendo datos a esta red. Esta clase recibe como parámetro una dirección de Internet a la cual el dispositivo se va a conectar y devuelve una cadena de caracteres indicando si la conexión fue exitosa o no.

- *PINEntry:* Esta clase envía instrucciones a la unidad central de procesamiento para que guarde la información proveniente del teclado que se refiere a la clave del usuario y que en pantalla se vean asteriscos. Esta clase recibe como parámetros un entero que indica el número de dígitos que se van a ingresar por el teclado, un entero que indica cual llave de cifrado debe utilizarse para cifrar los datos que se ingresen por el teclado, y un objeto tipo "Forma" que es la pantalla a mostrar mientras se digita la clave, que no muestra los números

digitados sino asteriscos. Al final, devuelve una cadena de caracteres indicando que la operación fue realizada satisfactoriamente.

- *ReadCardRequest:* Esta clase permite el envío de instrucciones a la unidad central de procesamiento para que reciba la información del lector de banda magnética. Esta clase recibe dos parámetros de tipo cadena de caracteres los cuales indican qué información recoger de la tarjeta deslizada: la Pista I o la Pista II. El primer parámetro hace referencia a la Pista I y el segundo parámetro hace referencia a la Pista II. Devuelve a la aplicación principal una cadena de caracteres indicando si la acción realizada fue correcta o no.

- *Reset:* Esta clase reinicia la unidad central de procesamiento. Envía al Wi – POS una instrucción que hace que el equipo reinicie. Se utiliza en algunos puntos de la aplicación principal para verificar que el funcionamiento es correcto y que no hay pérdida de información, o en procesos que pueden bloquear a la unidad central de procesamiento. Esta clase no recibe ningún parámetro de entrada ni devuelve algún dato.

- *Tabla:* Esta clase realiza la conversión de números en formato hexadecimal a enteros. Recibe como parámetro de entrada un carácter en hexadecimal, lo compara con una tabla que tiene y devuelve un entero, correspondiente al dato en hexadecimal.

- *Tramas:* Esta clase solicita a la unidad de procesamiento las tramas respectivas para enviar a Internet, espera una respuesta y devuelve a la unidad de procesamiento una trama. La clase recibe como parámetros de entrada tres datos de tipo entero. El primero hace referencia a la instrucción enviada al Wi – POS. El segundo hace referencia a la cantidad de bytes que se espera recibir del Wi – POS, dependiendo de la instrucción solicitada. El tercero indica el tamaño de la trama que devuelve el servidor en cantidad de caracteres, para guardarlos en una matriz, convertirlos a enteros para luego pasarlos a

hexadecimal y enviarlos byte por byte al Wi – POS.

- *Modulo10:* Esta clase verifica si el número de la tarjeta es un número válido haciendo uso del algoritmo de chequeo de modulo 10(ver Anexo Marco Teórico). Recibe como único parámetro una cadena de caracteres con el PAN (*Personal Account Number*) y devuelve "verdadero" (*true*) si el número de la tarjeta es un número válido, o "falso" (*false*) si éste es inválido.

Con estas clases, las aplicaciones principales pueden mantener contacto permanente con la unidad central de procesamiento y con el usuario/funcionario a través de la pantalla del teléfono.

4.3.3. Descripción de las aplicaciones principales.

A continuación se explicará la secuencia que siguen las aplicaciones de Carga de Parámetros Iniciales y de Transacción de Pago ilustrando lo que se ve en la pantalla del teléfono móvil utilizando el emulador del equipo Motorola iDEN i85s.

- Carga de Parámetros Iniciales

Los parámetros iniciales se solicitan al servidor, el cual devuelve una trama con ciertos parámetros que dependen del terminal que los solicita(Ver sección 3.3.1). Esta trama se almacena en la unidad central de procesamiento y al momento de la transacción se validan los diferentes datos con estos parámetros. La secuencia de esta aplicación se da de la siguiente forma:

Pantalla Inicial

En esta pantalla se le informa al funcionario que puede cargar los Parámetros Iniciales.

Figura 21. Pantalla Inicial

Figura 22. Carga de parámetros en proceso

Carga de Parámetros en proceso

En esta pantalla se le informa al funcionario que se está efectuando la carga de los Parámetros Iniciales en el equipo. Durante la carga de los parámetros se despliega una barra que muestra el progreso del proceso. En este momento la aplicación se comunica con el dispositivo y le solicita la trama de Solicitud de Parámetros Iniciales, la cual es enviada al servidor. El servidor devuelve una trama de respuesta con todos los parámetros iniciales, que recibe la aplicación y que inmediatamente es enviada al dispositivo.

Figura 23. Carga de Parámetros Iniciales
Carga de parámetros correcta.

Confirmación de la carga de los parámetros

Se le informa al usuario que la carga de parámetros iniciales ha sido exitosa y por lo tanto ya se pueden realizar transacciones de pago.

Figura2 4,
Carga de parámetros incorrecta

Carga de parámetros incorrecta: En esta pantalla se le informa al usuario que la solicitud y carga de parámetros iniciales ha tenido problemas. Este error ocurre cuando la trama de respuesta enviada por el servidor al dispositivo no es la esperada y hay información errónea en algún campo de la trama.

Figura 25. Carga de Parámetros Iniciales,
Error en el puerto serial

Error en la comunicación con Wi – POS

Si durante el proceso de carga de parámetros se da algún error en la comunicación entre el teléfono móvil y Wi – POS, se le informa al funcionario de este error para que reinicie la aplicación. Cuando el funcionario oprime "OK", la aplicación automáticamente vuelve a comenzar, desplegando la pantalla inicial, mostrada anteriormente. La aplicación intenta realizar la carga de parámetros hasta tres veces si hay inconvenientes en la comunicación con Wi – POS. Después de la tercera vez, se le informa al usuario que la carga de parámetros no es posible en el momento. Estos errores pueden ocurrir si el dispositivo no está conectado de la forma correcta al teléfono móvil.

Figura 26. Carga de Parámetros Iniciales,
Error en la conexión a Internet

Error en la conexión a Internet

Si durante el proceso de carga de parámetros se da algún error en la comunicación entre el teléfono móvil y el servidor de Internet, se le informa al funcionario de este error para que reinicie la aplicación. Cuando el funcionario oprime "OK", la aplicación automáticamente vuelve a comenzar, desplegando la pantalla inicial, mostrada anteriormente. La aplicación intenta realizar la carga de parámetros hasta tres veces si hay inconvenientes en la comunicación con el servidor. Después de la tercera vez, se le

informa al usuario que la carga de parámetros no es posible en el momento. Estos errores se pueden presentar si la dirección a la que intenta acceder la aplicación no es válida o si el servidor en ese momento no está en servicio.

Figura 27. Carga de Parámetros Iniciales, Error sin solución

No se puede solucionar el error

Esta pantalla aparece cuando se ha intentado realizar la carga de parámetros hasta tres veces y persiste el error de comunicación con Wi – POS o el error de comunicación con Internet. En este caso la aplicación le informa al funcionario que debe salir de la aplicación para intentar de nuevo todo el proceso.

- . Transacción de Pago

Esta aplicación es la que permite al usuario realizar una transacción de pago utilizando tarjeta débito o tarjeta de crédito. El proceso de la transacción es descrito a continuación mostrando cada pantalla que despliega la aplicación, de nuevo utilizando el emulador del equipo móvil Motorola iDEN i85s. Es necesario que los parámetros iniciales estén cargados en el dispositivo antes de poder realizar la transacción de pago. Por esta razón hay que ejecutar primero la aplicación de carga de Parámetros Iniciales para luego poder efectuar con éxito la transacción de pago.

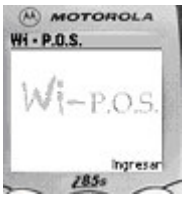


Figura 28. Transacción de Pago, Pantalla Inicial

Pantalla Inicial

Esta aplicación comienza con una pantalla de inicio, desplegando el logotipo de Wi – POS. Cuando el usuario oprime “Ingresar” lo primero que hace la aplicación es verificar que los Parámetros Iniciales estén cargados en el

dispositivo Wi – POS. La aplicación le envía una instrucción al dispositivo para que le responda si los parámetros están cargados. Si no están cargados, se despliega una pantalla informando al usuario que debe cargar los Parámetros Iniciales antes de intentar realizar la transacción pero si éstos ya están cargados, la aplicación sigue su curso.

Figura 29. Transacción de Pago, Parámetros Iniciales no han sido cargados

Parámetros Iniciales no están cargados

Si los Parámetros Iniciales no han sido cargados, se le pide al usuario que salga de la aplicación y los cargue de inmediato.

Figura 30. Transacción de Pago, Pantalla de bienvenida

Pantalla de Bienvenida

En esta pantalla se le da la bienvenida al usuario y se pide que verifique el dispositivo está correctamente conectado el equipo móvil Motorola.

Figura 31 Transacción de Pago, Deslizar tarjeta

Deslizar tarjeta crédito/débito por el lector

En esta pantalla se le pide al usuario que deslice la tarjeta de crédito/débito por el lector de banda magnética. Si la tarjeta es deslizada incorrectamente aparece una pantalla informándole al usuario que la deslizó de forma indebida y que vuelva a intentarlo. De forma contraria, la transacción sigue su curso.

Figura 32, Error de tarjeta

Figura 33. Transacción de Pago, Verificando Datos

Verificación de datos

En esta pantalla se le pide al usuario que espere un momento mientras se verifican algunos datos. En este instante la aplicación le pide al dispositivo que le envíe la trama de Solicitud de los Parámetros de Tarjeta para enviarla al servidor y que éste le devuelva una trama con dichos parámetros para almacenarlos en el dispositivo y realizar el proceso de la transacción según lo indiquen los parámetros de tarjeta. Al finalizar la carga de los parámetros de tarjeta, la aplicación verifica si el parámetro "módulo 10" está activo y si lo está, procede a ejecutar la clase Modulo10 para verificar que el número de la tarjeta deslizada por el lector es válido.

En caso que sea válido, aparece en pantalla la opción "Continuar" para que el usuario prosiga con la transacción. Si el número no es válido, aparece una pantalla indicándole al usuario de este problema y tiene tres intentos para pasar por el lector una tarjeta válida. Si pasan los tres intentos y la tarjeta no es válida, la aplicación le informa al usuario que debe abandonar la transacción. Si existe algún error de comunicación entre la aplicación y el dispositivo o entre la aplicación y el servidor,

aparecerá la respectiva pantalla indicando dicho error, así como en la aplicación de Parámetros Iniciales.

La pantalla mostrada en la figura 34 , se muestra mientras el equipo se comunica con el servidor y le solicita los parámetros de la tarjeta. Estos parámetros son cargados y solicitados, además en este momento se hace una solicitud de llave de cifrado de PIN de ser necesario y se valida el modulo 10 de la tarjeta.

Figura 34. Verificación de datos

Esta pantalla aparece cuando el número de la tarjeta es inválido y ésto lo determina la clase Modulo10. Hay tres oportunidades para que el usuario deslice una tarjeta válida por el lector de banda magnética.

Figura 35, Número inválido

Si después de los tres intentos el usuario no desliza una tarjeta válida, la transacción termina y se le informa al usuario que la aplicación debe terminar.

Figura 36, Persiste error de Número Invalido

Una vez cargados los parámetros de tarjeta en el dispositivo, la aplicación le pide uno a uno el valor de los diferentes parámetros a Wi - POS para almacenarlos en variables que la aplicación consulta para pedir los diferentes datos al usuario a medida que transcurre el proceso de la transacción. Si alguno de los parámetros no está activo para solicitarlo, la aplicación prosigue a verificar el siguiente. A continuación se muestran cada uno de los parámetros que se solicitan en el orden en que la aplicación los requiere.

El primer parámetro a validar es la fecha de vencimiento. Si este parámetro está activo, se le solicita al usuario ingresar la fecha de vencimiento

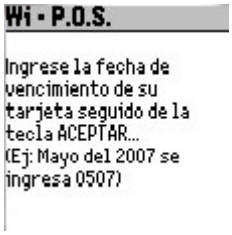


Figura 37, Ingrese fecha

Si la fecha de vencimiento ingresada por el usuario no corresponde a la de la tarjeta, se le informa que la fecha es inválida y que debe salir de la aplicación.

Figura 38, Fecha Inválida

El siguiente parámetro es comparar la fecha de la tarjeta con la fecha del sistema. Si está activo, internamente el dispositivo compara las dos fechas y si la tarjeta está vigente continúa la transacción, si no, aparece esta pantalla.

Figura 39, Tarjeta Vencida.

Los siguientes parámetros a verificar indican si la aplicación debe pedir al usuario el ingreso de una clave, ya sea para tarjeta débito o para tarjeta de crédito. Si estos parámetros están activos, aparece la pantalla que le pide la clave al usuario.

Figura 40. Ingreso Clave

A continuación se le pide al usuario el tipo de tarjeta, si este parámetro se encuentra activo. El usuario escoge entre tres opciones: tarjeta débito de ahorros, tarjeta débito corriente o tarjeta de crédito.

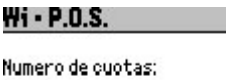
Figura 41, Selección de tarjeta

En este momento se le pide al usuario que ingrese el valor de la transacción utilizando el teclado de Wi – POS.

Figura 42, Valor Transacción

El siguiente parámetro a verificar es si se necesita pedir propina al usuario. Si esto es necesario, el usuario ingresa el valor de la propina utilizando el teclado del Wi – POS.

Figura 43, Valor de Propina



A continuación se verifica si es necesario pedirle al usuario el número de cuotas a diferir el pago a realizar. Si está activo, el usuario debe ingresar el número de cuotas por el teclado del dispositivo.

Figura 44, Número de cuotas

El último paso es la validación de la transacción. Una vez solicitado el último parámetro, la aplicación calcula el IVA del valor introducido, la Base de Devolución de IVA (Valor de la transacción sin IVA) y los envía al Wi – POS. Luego solicita al dispositivo la Trama de Pago para enviarla al servidor. El servidor devuelve una trama con la descripción de la respuesta a la transacción. Mientras este proceso ocurre, se despliega una pantalla de Transacción en Proceso.

Figura 45, Transacción en proceso

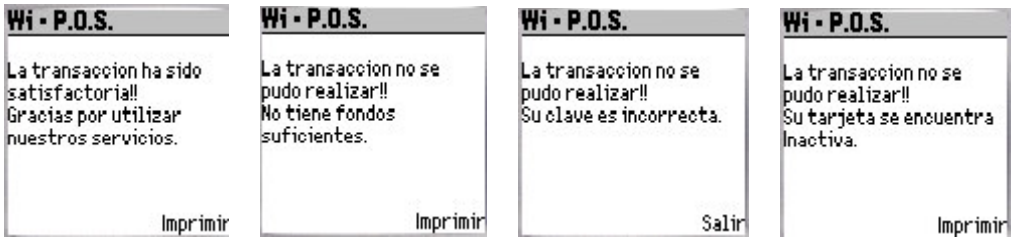


Figura 46, Pantallas de resultado de transacción

Dependiendo del resultado de la transacción se desplegara una pantalla informando la aprobación o error de esta, alguno casos se muestran en la figura 45

Esto lo realiza el teléfono cuando por medio de la aplicación y después de haber recibido la repuesta de transacción, pide al microcontrolador los campos de código de repuesta y descripción de respuesta(Ver sección 3.3.6)

Luego de obtener el resultado de la transacción, se imprime un comprobante de pago solamente si el parámetro que lo solicita está activo. En caso contrario, no se imprime ningún comprobante. El comprobante incluye el valor de la transacción, el consecutivo y el numero de terminal en donde se realizo esta.

4.4. APLICACIÓN EN VISUAL BASIC

La finalidad de esta aplicación desarrollada en Visual Basic es ingresar la

información de las llaves maestras, de la llave de transporte y del número de terminal. Esta aplicación fue implementada externamente al teléfono móvil para que no todos los usuarios de WI-P.O.S tengan la posibilidad de acceder a estas funciones, es por esto que se recurrió a una aplicación desarrollada en Visual Basic. Para su uso es necesario tener conectado WI-P.O.S por medio de un cable serial RS232 a un computador. Para la comunicación se usaron instrucciones de 8 bits. A continuación en la Figura 47, se muestra la pantalla principal de la aplicación.

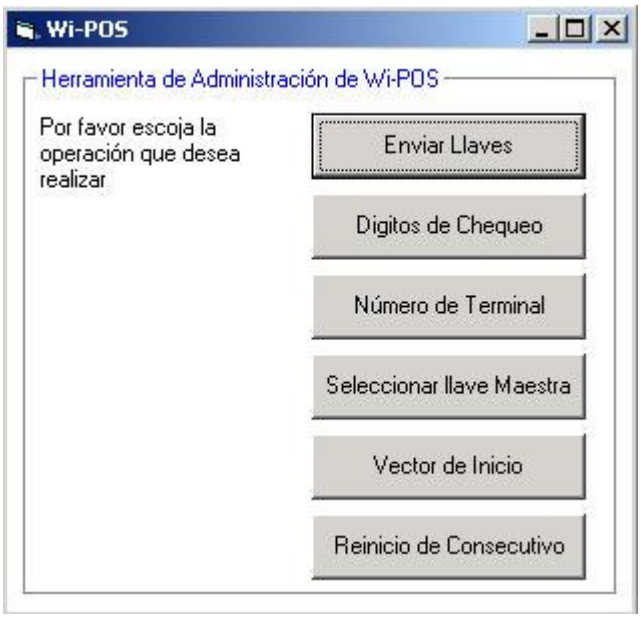


FIGURA 47, Pantalla principal de la Herramienta de Administración de WI-P.O.S

A partir de las opciones mostradas en el menú principal de la aplicación se explicarán las diferentes funciones del programa.

4.4.1. Enviar Llaves

Esta función sirve para enviar las Llaves Maestras o de Transporte al equipo (Para ver información de Llaves Maestras y de Transporte Ver Capítulo de Especificaciones, numeral 3.4). A continuación se observa la Figura 48, donde se muestran las diferentes opciones de la sección Enviar Llave.

FIGURA 48, Pantalla de Enviar Llaves.

En esta sección se tienen 2 opciones que son las de enviar la llave de transporte o la llave maestra.

4.4.2. Llave de Transporte

En la Figura 49, se muestra la pantalla de la aplicación desde donde se envía la Llave de Transporte.

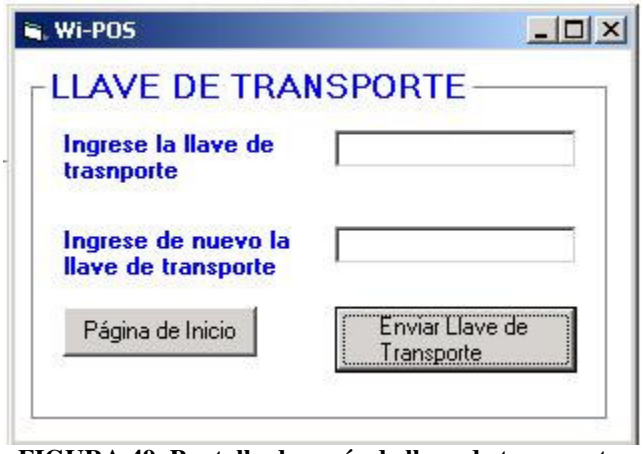


FIGURA 49, Pantalla de envío de llave de transporte.

Para enviar la llave de transporte a WI-P.O.S se manejó un protocolo de comunicación entre las dos partes, todo transmitido y recibido desde el puerto serial del computador. El protocolo establecido para estos procesos esta ilustrado en la Figura 50.

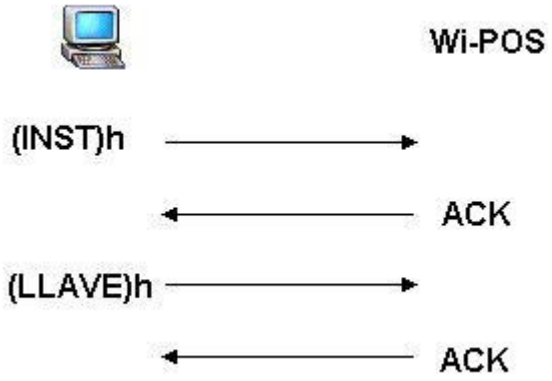


FIGURA 50. Protocolo de envío de llave de transporte

La estructura de la aplicación consiste en primero validar la información de la llave de transporte, por lo que se le pide al usuario que ingrese la llave de transporte 2 veces, estas dos llaves deben ser las mismas en longitud y valor, en el momento que se ingresen llaves diferentes o la longitud de las llaves sean diferentes a 16 caracteres la aplicación avisará que se ha producido un error y no enviará la información como se puede ver en la Figura 51 y luego pedirá que se ingresen de nuevo. Al ingresarse las llaves correctamente, la aplicación envía primero a WI-P.O.S la instrucción correspondiente a la carga de la llave de transporte, WI-P.O.S realiza un reconocimiento de la información recibida y envía un ACK para que la aplicación le envíe la llave de transporte.

FIGURA 51, Errores producidos al ingresar mal los datos de llave de transporte

Como la información ingresada de la llave se encuentra en ASCII, la aplicación la transforma en Hexadecimal, tomando el entero correspondiente al carácter en ASCII

y restándole 48 en caso de ser un valor entre 0 y 9, 55 en caso de ser un valor entre "A" y "F", o 87 en caso de haber escrito los valores de "a" a "f" en minúscula.

Luego que WI-P.O.S recibe la llave de transporte este procede a enviarle otro reconocimiento es decir un ACK, con este último valor la aplicación en Visual sabe que la información fue enviada satisfactoriamente y avisa al usuario del éxito del proceso, como se puede ver en la Figura 52.

FIGURA 52, Envío de la llave de transporte exitosa.

Cabe anotar que en caso que se tenga problemas en la comunicación de WI-P.O.S con el computador, la aplicación avisará al usuario para verificar la conexión como se puede ver en la Figura 53.

FIGURA 53, Error en la comunicación entre WI-P.O.S y el Computador.

4.4.3. Llave Maestra

La función de ingreso de la llave maestra varía con respecto a la función de la llave

de transporte en que se deben seleccionar el número total de subllaves maestras y el número de la subllave que se va a enviar. En la Figura 54, se puede ver la pantalla en que se selecciona el número de subllaves totales.

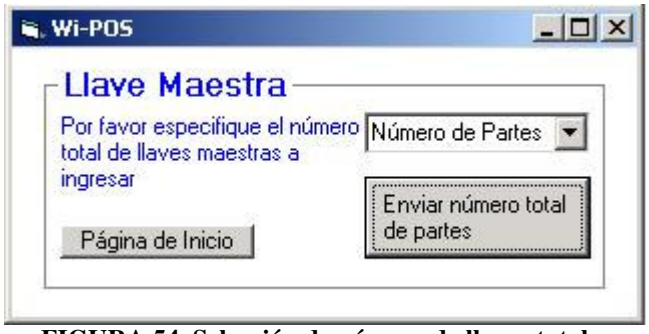


FIGURA 54, Selección de número de llaves totales

Para enviar el número total de llaves maestras a enviar se le envía por el puerto serial a WI-P.O.S la instrucción correspondiente a la rutina para que acepte estos valores, es importante almacenar éste valor porque determina entre cuantas partes se debe hacer XOR, por lo que es almacenado en la EEPROM. En la Figura 55 se observa como se maneja la instrucción de envío de número de partes.

FIGURA 55, Envío de número total de llaves maestras

En el momento que se ha aceptado el número de partes la aplicación de Visual Basic despliega la pantalla en que se envía la llave maestra, como se ve en la Figura 56.

FIGURA 56, Envío de llave maestra

Para enviar la información de la llave maestra se envía primero la instrucción a WI-P.O.S para que entre en la rutina de ingreso de llave maestra, luego se envía el número de la parte y finalmente la llave maestra. Cabe anotar que se hacen las mismas validaciones de tamaño de la llave e igualdad entre la llave maestra y la confirmación de la llave maestra como se explicó en la sección de llave de transporte. En la Figura 57 se ilustra el proceso de comunicación para enviar las llaves maestras.

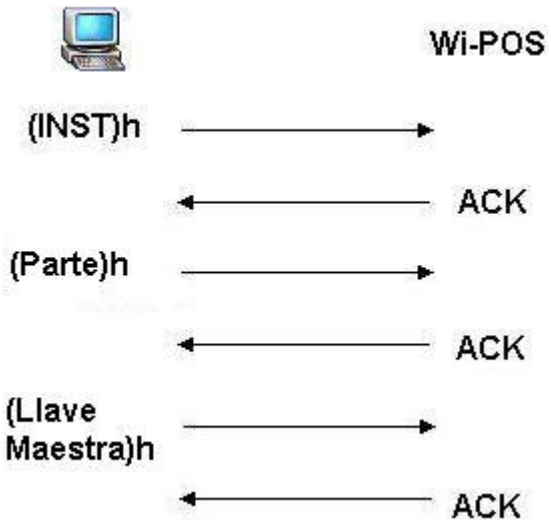


FIGURA 57. Comunicación entre WI-P.O.S y Computador para envío de Llaves Maestras

4.4.4. Seleccionar Llave Maestra

Como se puede ver en el Capítulo de Especificaciones, en la sección correspondiente a llaves maestras, en el equipo se pueden almacenar 3 tipos de llaves para tener un

mejor manejo de la seguridad. A continuación se ilustra en la Figura 58, la pantalla de la aplicación en Visual Basic en donde se realiza la selección de la llave maestra.

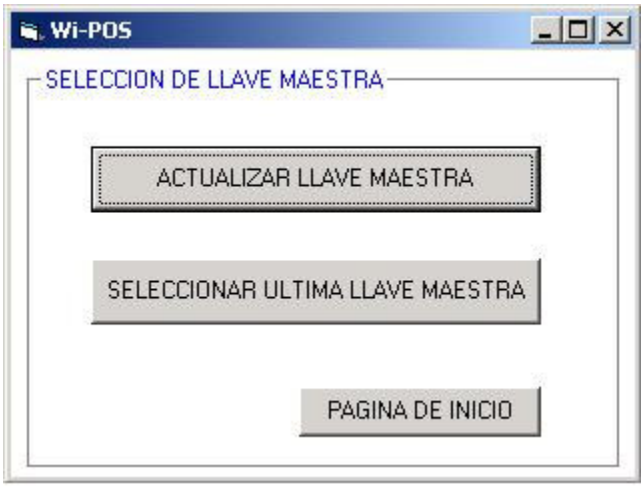


FIGURA 58. Selección de llave maestra

Al igual que en los anteriores procesos de comunicación, WI-P.O.S recibe las correspondientes instrucciones que le permiten realizar los procesos de cambios de llave. Al realizarse la operación satisfactoriamente la aplicación avisa al usuario como se puede ver en la figura 59.

FIGURA 59. confirmación de actualización del programa

4.4.5. Dígitos de Chequeo

Los dígitos de chequeo sirven para saber si una llave fue ingresada correctamente en WI-P.O.S, estos dígitos corresponden a los cuatro primeros caracteres resultantes de cifrar ceros con la llave ingresada, de esta manera puede comparar estos valores con unos previamente asignados para la verificación del correcto ingreso de la llave. A continuación en la figura 60 se puede observar la ventana en donde se verifica los dígitos de chequeo de la llave de transporte y de la llave maestra.

En el momento que se oprime el botón de chequeo, la aplicación envía la instrucción correspondiente para que WI-P.O.S genere los dígitos de chequeo y los envíe al computador. En la aplicación de Visual Basic la información llega en formato hexadecimal, por lo tanto para que aparezca en la ventana se debe transformar a formato ASCII de la siguiente manera, si recibe valores entre 0 y 9 la aplicación le sumará 48 y si llegan valores entre A y F la aplicación le sumará 55. De esta manera se tiene el valor correspondiente en ASCII.

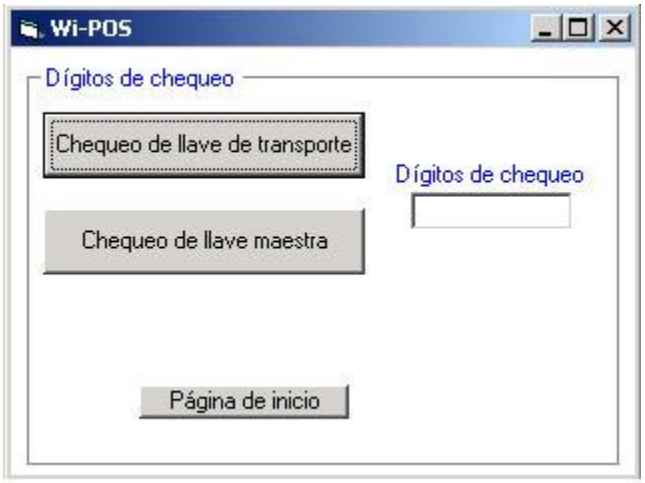


FIGURA 60. Ventana de dígitos de chequeo

4.4.6. Número de terminal

Para que en una transacción bancaria se identifique a quien realiza la operación, el sistema debe identificar el número de terminal que realiza la transacción, es por esto

que cada WI-P.O.S debe tener un número único que lo identifique. Por medio de esta sección de la aplicación se pueden ingresar a WI-P.O.S estos valores, en la siguiente figura se observa la ventana donde se ingresa el terminal, cabe anotar que puede tomar valores hasta de 8 caracteres.

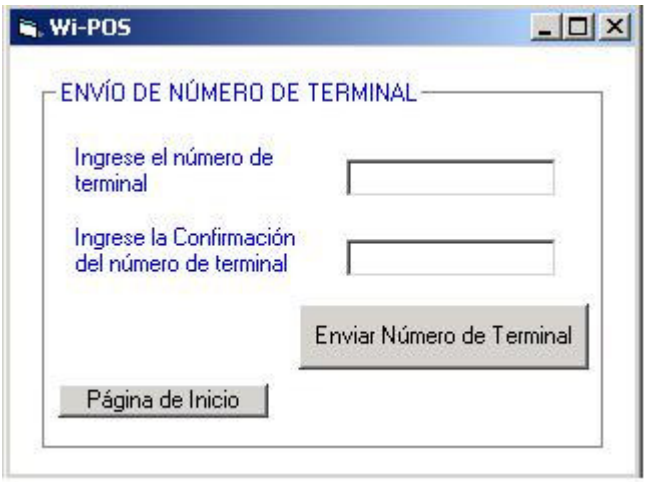


FIGURA 61, ventana de envío de número de terminal

Al igual que en el envío de la llave de transporte, el sistema valida los caracteres en tamaño y en igualdad antes de enviarlos, de tal manera que avisa al usuario si encuentra alguna inconsistencia.

4.4.7. Vector de Inicio

El vector de inicio es un parámetro necesario para la realización de cifrado en DES en modo CBC (Ver Sección 4.1.5) y es necesario que esté ubicado en Wi-P.O.S. y en el servidor donde llegan las tramas de solicitud para que el cifrado y descifrado sean correcto. Para su implementación se utilizó una interfaz al igual que la llave de transporte donde se debe escribir el valor del vector de inicio y su confirmación por cuestiones de seguridad en el ingreso de la información, a continuación se muestra la pantalla donde se ingresan los datos.

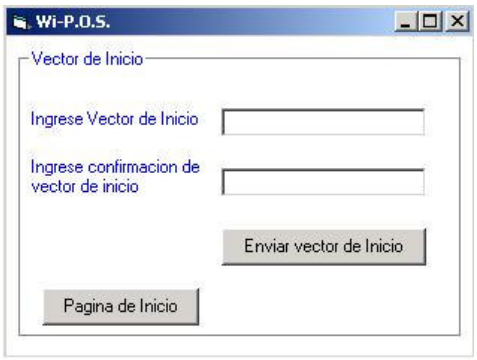


FIGURA 62, ventana de envío de vector de inicio

En esta interfaz se maneja un protocolo de comunicación similar al de la carga de llave de transporte, enviando la instrucción para que Wi-P.O.S. entre a rutina de recolección de vector de inicio, luego recibe reconocimientos y envía la información correspondiente al vector de inicio.

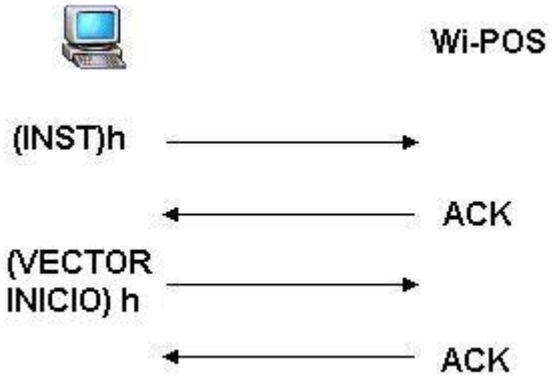


FIGURA 63. Comunicación entre WI-P.O.S y Computador para envío de vector de inicio

4.4.8. Inicio de consecutivo

El consecutivo es un valor que identifica la transacción realizada por el terminal específico, es decir, no se van encontrar 2 transacciones hechas por un mismo Wi-P.O.S. que tengan un mismo consecutivo, estos valores son puestos en 000000 a partir de una instrucción enviada desde el administrador de Wi-P.O.S., a su vez para la confirmación del procedimiento de reinicio de consecutivo el administrador espera una instrucción de confirmación. Para este procedimiento se utilizó la siguiente

interfaz en donde solo se envía la instrucción y se espera por la confirmación. (Ver Figura 64)

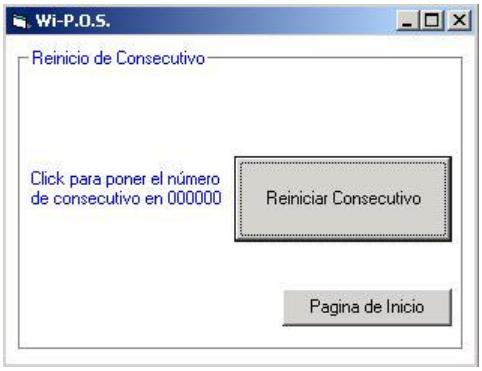


FIGURA 64, ventana de envío de vector de inicio

4.5 DISEÑO DEL CIRCUITO IMPRESO

Se diseñó un impreso que incluyera en una sola tarjeta el microcontrolador y sus periféricos y la FPGA ACEX con su memoria serial, para su diseño se tuvieron en cuenta los siguientes aspectos.

- El tamaño debía ser lo más reducido posible para que el producto final sea cómodo de transportar.
- Se incluyeron dos reguladores, el primero de 2.5V para alimentar ciertos pines de la ACEX. El segundo de 3.3V para alimentar el resto de integrados.
- La distribución de las fuentes se realizó en estrella, es decir se envían caminos hacia puntos centrales en donde se ramifican los caminos.
- Se colocaron condensadores de 100nf cerca de los integrados. Estos para desacople.
- Las conexiones entre el microcontrolador y la ACEX (módulo de cifrado) se hicieron con caminos cortos y rectos. Esto se logró seleccionando los pines indicados tanto en el microcontrolador como en la FPGA y ubicando de manera apropiada los componentes.

- Se usaron componentes de montaje superficial, para reducir el tamaño.
- Se hizo plano de tierra para disminuir interferencias, ya que se trabaja cerca de un teléfono móvil.
- Lograr una distribución adecuada de los componentes para evitar caminos largos y enredados.

A continuación se muestra una fotografía del circuito impreso diseñado.

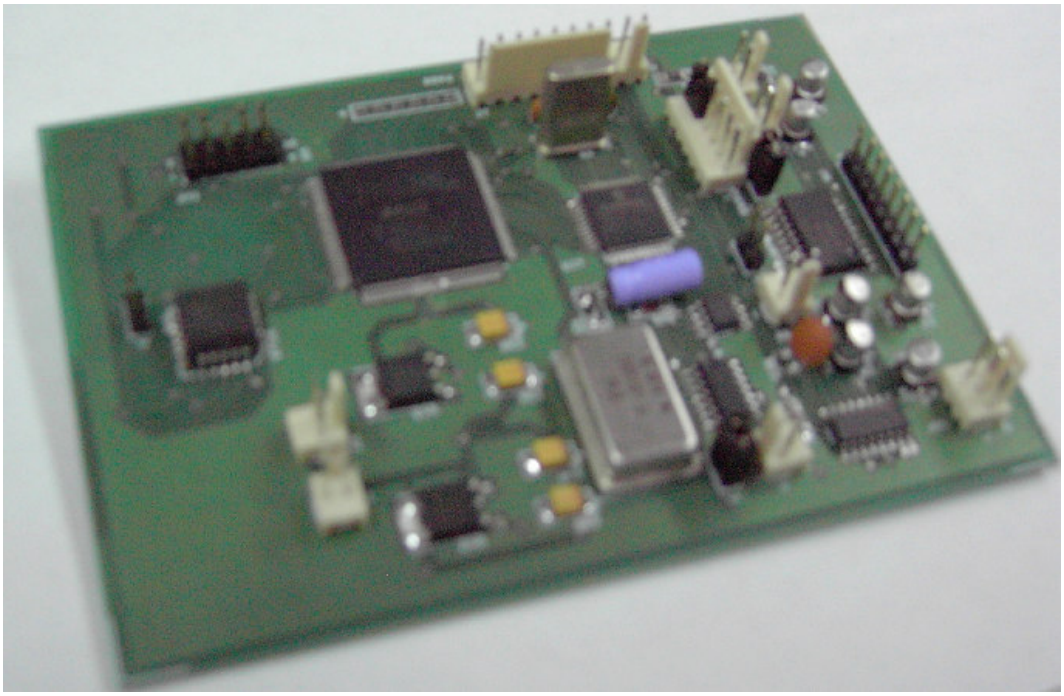


Figura 65, Fotografía del circuito impreso

5. PRUEBAS Y ANÁLISIS DE RESULTADOS

5.1 VERIFICACIÓN DE FUNCIONAMIENTO DEL MÓDULO DE CIFRADO.

Para verificar el correcto funcionamiento del módulo de cifrado se utilizó un servidor con tarjeta criptográfica y se diseñó una rutina en la cual se le envía la llave de cifrado y los datos a cifrar desde un computador al microcontrolador y este a su vez se comunica con el módulo de cifrado (este procedimiento se explica en las secciones 4.1.5 y 4.2), luego espera la respuesta y la envía al computador. Esta prueba se realizó en el circuito impreso en el cual ya se encuentran interconectados el microcontrolador y el módulo de la manera adecuada, de modo que lo que se hizo fue tomar varias llaves y datos que luego se cifraron usando la tarjeta criptográfica y el módulo desarrollado en el presente trabajo, luego se compararon los resultados. Para el uso de la tarjeta criptográfica se contó con la asesoría de un ingeniero asociado a la empresa ASIC que tiene acceso al servidor con tarjeta criptográfica y conocimiento del manejo de esta.

A continuación se describen las pruebas realizadas.

5.1.1 Pruebas de verificación de cifrado/descifrado de un bloque sin vector de inicio:

Se realizaron pruebas comparando la información cifrada resultante en WI-P.O.S. con la información cifrada por la tarjeta criptográfica, estos resultados se muestran en el Anexo 2, en la sección de pruebas de cifrado.

De esta misma forma se realizaron otras pruebas similares de resultado satisfactorio que incluyen en su totalidad en el documento, comprobando que el módulo de cifrado funciona de la manera adecuada, ya que no se presentaron errores.

5.1.2 Pruebas de verificación de funcionamiento de cifrado en modo CBC.

Como se explicó en el capítulo 2 el módulo de cifrado sólo cifra bloques de 64 bits, por tanto para realizar cifrado en modo CBC se elaboró una rutina del microcontrolador (ver sección 4.1.5) que lleva a cabo el proceso necesario. Para probar el buen funcionamiento de ésta, se utilizó el servidor con tarjeta criptográfica, y de manera similar a la descrita en la sección anterior, se hicieron pruebas y se compararon resultados. Algunas de estas pruebas se muestran en detalle en el anexo 2, en los resultados de funcionamiento DES en modo CBC

Las pruebas realizadas tuvieron resultados satisfactorios . Esta serie de pruebas demostró que el módulo de cifrado y su interconexión con el microcontrolador funcionan correctamente, cifrando los datos de la manera adecuada en modo CBC.

5.2 PRUEBAS DE LA APLICACIÓN REALIZADA EN VISUAL BASIC

La aplicación realizada en Visual Basic es una interfaz entre el usuario y Wi-P.O.S. para poder introducir las configuraciones específicas del dispositivo (Para mayor información remitirse a la sección 4.4.).

Para la realización de las pruebas se implementaron los siguientes procesos para determinar si los parámetros fueron correctamente introducidos al dispositivo. En cada uno de ellos se debía tener conectado Wi-P.O.S. a un computador en donde se encontraba cargada la aplicación. La conexión se realiza al puerto serial del computador.

5.2.1.Prueba de carga de llave de transporte y verificación por dígitos de chequeo.

Para probar el correcto funcionamiento de la introducción de la llave de transporte se utilizó la función implementada de dígitos de chequeo, la cual se encarga de devolver los 4 primeros caracteres resultantes de cifrar una cadena de ceros con la llave

elegida.
Para asegurar que el resultado fuese correcto se cifro una cadena de ceros con la misma llave en el servidor de tarjeta criptográfica y se compararon los 4 primeros caracteres resultantes con los obtenidos en el proceso de Dígitos de Chequeo.

El proceso en el administrador de Wi-P.O.S. es el siguiente: Se escoge la opción "Enviar Llaves", luego "Llave de transporte", luego se introduce la llave y se envía. Al volver al menú principal se escoge la opción "dígitos de chequeo", donde se selecciona la opción correspondiente a "Llave de transporte" y de esta manera se obtiene la información a comparar con el resultado de la tarjeta criptográfica. (Ver sección 4.4). Algunas de las pruebas se muestran en el Anexo 2, sección de pruebas de funcionamiento de carga de llaves de transporte.

Como se puede ver en los resultados, los dígitos de chequeo coinciden con los 4 primeros caracteres del dato cifrado en todas las pruebas. Esto demuestra que la carga de llave de transporte se hace satisfactoriamente.

5.2.2. Prueba de la carga de llaves maestras y verificación por dígitos de chequeo.

Esta prueba fue similar a la realizada con la llave de transporte, su única diferencia es que la llave maestra debe ser introducida por partes como se describe en las secciones 1.4.1 y 2.4 , teniendo en cuenta que la llave maestra final es el resultado de la operación XOR entre todas éstas. El proceso de prueba varía en que antes de solicitar los dígitos de chequeo se debe actualizar la llave maestra escogiendo la opción de "Selección de llave maestra" en el menú principal. Los resultados de algunas de las pruebas se describen en el Anexo 2, sección de pruebas de funcionamiento de carga de llaves maestras.

De esta manera se comprobó el funcionamiento satisfactorio de la carga de llaves maestras en el equipo haciendo uso de la aplicación en Visual Basic

5.2.3. Prueba de Selección de llave maestra.

Para la realización de las pruebas se utilizaron los resultados obtenidos en el numeral 3.2.2. La prueba consistía en cargar una llave nueva al sistema, comparar los dígitos de chequeo y luego cargar la anterior llave maestra de la cual también se obtenían los dígitos de chequeo. Para una mayor información acerca de los resultados, favor remítase al anexo 2, sección de selección de llave maestra.

5.2.4. Prueba de carga de número de terminal

En el "banco virtual" (Ver sección 5.3) se establecen parámetros iniciales(ver sección 3.3) de acuerdo al número de terminal. La prueba de verificación de la introducción de número de terminal usando la herramienta en Visual Basic consistió en tener 3 tipos de terminales diferentes cada uno con parámetros propios, como por ejemplo, llave de PIN de diferente longitud, valor de IVA y solicitud de propina. Esta información se puede ver más claramente en la TABLA II donde se muestran los valores correspondientes. Esta prueba se encuentra ligada a la sección 5.3.1 ya que las tramas enviadas involucran el número de terminal.

5.2.5. Prueba de la introducción de vector de inicio:

Para verificar el funcionamiento de la introducción de vector de inicio se configuró un valor para el vector de inicio en el "Banco virtual", (Ver sección 5.3) que fue también introducido por el administrador de Wi-P.O.S., las pruebas se pueden consultar en cada uno de las solicitudes hechas en el numeral 5.3.

5.2.6. Prueba de Reinicio de número de consecutivo

Reiniciar el consecutivo significa poner seis ceros en el registro de memoria donde se

almacena el consecutivo. Como el consecutivo es uno de los parámetro enviados al servidor al momento de realizar una transacción de pago, las pruebas de reinicio de número de consecutivo van ligadas a las pruebas de realización de la transacción de pago descritas en la sección 5.3.2.

5.3. PRUEBAS DE FUNCIONAMIENTO DEL DISPOSITIVO REALIZANDO TRANSACCIONES BANCARIAS:

Debido a que el Wi-POS es un prototipo y las políticas de seguridad de VISA sólo permiten el acceso a los proveedores tras un complejo proceso de certificación, se implementó un ambiente de pruebas en donde se simula un banco y tiene las siguientes características:

- Un servidor con IP pública para tener acceso desde el teléfono Motorola.
- Una tarjeta criptográfica similar a la utilizada en los procesos de transacciones bancarias.
- Una base de datos almacenando información correspondiente a varios usuarios del banco virtual.
- Un software que se encarga de realizar la aprobación o rechazo de las transacciones.

Todo este ambiente se desarrolló bajo la asesoría de la empresa ASIC, que es una empresa que ejecuta proyectos con VISA y conoce los procesos de las transacciones bancarias. Es a través de una implementación de dicha entidad que una transacción se podría llevar a cabo con la mensajería que se envía desde Wi-POS (Versecciòn3.3)

5.3.1 Pruebas del proceso de carga de parámetros iniciales:

Como se explicó en la sección 3..3, los parámetros iniciales dependen del terminal que los requiere, por esta razón, para verificar que se esté efectuando correctamente el

proceso, se almacenó en la base de datos la información correspondiente a tres números de terminal diferentes con parámetros iniciales distintos y de esta manera se pudieron tener varias opciones en cada parámetro.

En el proceso se envía al servidor la trama denominada "petición de parámetros iniciales" debidamente cifrada, usando el modo CBC del algoritmo DES, con llave de transporte y vector de inicio comunes previamente ingresados en el dispositivo (Ver sección 4.4.2) y en el banco virtual.

El software instalado en el servidor "Banco Virtual", se encarga de descifrar la trama utilizando la tarjeta criptográfica, luego reconoce el tipo de mensaje (Ver sección 3.3) y si se trata de un mensaje de petición de parámetros iniciales, busca en una tabla de la base de datos la información correspondiente al terminal contenido en el mensaje. Con esta información construye una trama de respuesta de parámetros iniciales, luego la cifra y la envía como respuesta al teléfono móvil.

Los tres terminales ingresados tenían los siguientes datos:

TABLA II. Números de terminal consignados en la base de datos y sus parámetros correspondientes.

	Número de terminal	IVA	Propina	Dígitos de Clave	Base devolución de IVA	Emisor
1	12345678	16	Habilitado	4	Habilitado	VISACENTRO
2	87654321	12	Deshabilitado	4	Habilitado	UJaveriana
3	13482579	15	Habilitado	5	Deshabilitado	CENTROVISA

Para ver que el proceso se estuviera llevando a cabo adecuadamente, se desarrolló una página web en donde se podían ver los mensajes que entraban y salían del "Banco Virtual", cifrados y en claro.

En el anexo 2, sección de pruebas de funcionamiento de parámetros iniciales, se muestra en detalle lo sucedido en varias de estas pruebas. Con los resultados obtenidos en estas pruebas se comprobó el buen funcionamiento del equipo en lo que respecta a la construcción y cifrado/descifrado de las tramas y la conexión con el servidor a través de la red iDEN

5.3.2 Pruebas del proceso de transacción de pago con tarjeta crédito o debito.

El proceso completo de la transacción de pago depende de los parámetros iniciales cargados previamente y de la tarjeta deslizada. Como se vio en la sección anterior se dispuso de tres tipos de terminal con diferentes parámetros iniciales y además de esto, la base de datos contaba con un registro de usuarios del banco. Para esto, se usaron varias tarjetas crédito y débito, propias de los miembros del grupo. A cada tarjeta se le asignaron distintos parámetros para poder probar diferentes casos en las transacciones.

Para realizar este procedimiento, el banco virtual procesa tres tramas diferentes en distintos tiempos para poder validar la transacción, estas tramas son: Trama de solicitud de parámetros de tarjeta, trama de solicitud de llave de PIN y trama de solicitud de transacción de pago (Ver sección 3.3). En el proceso el servidor descifra cada una de estas tramas en el momento que las recibe, valida la información, genera respuestas dependiendo de cada solicitud y finalmente cifra la información para enviarla de nuevo a Wi-POS. Para ver el proceso completo de una transacción el lector puede remitirse al diagrama de flujo 3. Con el banco virtual se puede verificar si el proceso se lleva a cabo satisfactoriamente, si los campos que se le piden al usuario a través de la pantalla del teléfono móvil, son acordes a los parámetros enviados. Para ver la secuencia de pantallas del teléfono móvil, que se debe observar en el proceso de la transacción remítase a la sección 4.3.3. Para observar los resultados de las pruebas de funcionamiento, remítase a las pruebas de funcionamiento de transacciones en el anexo 2.

5.3.3. Pruebas con los posibles errores de una transacción de pago.

Así como se hicieron pruebas de transacciones satisfactorias se evaluó el funcionamiento del dispositivo ante errores con los siguientes casos:

- El usuario desliza mal la tarjeta: Caso en el cual el teléfono debe mostrar en pantalla un mensaje de error al leer tarjeta y mostrar al usuario la posibilidad de salir de la aplicación o de continuar.
- Fecha de vencimiento incorrecta: En el caso de ser solicitada, el usuario ingresa de forma errónea la fecha.
- Clave incorrecta: Se digita incorrectamente la clave. Por tanto la transacción no se puede realizar y debe mostrarse una pantalla que lo indique.
- Tipo de cuenta errónea. En el momento en que se muestra la pantalla de seleccionar el tipo de cuenta el usuario elige una cuenta que no tiene registrada en la basa de datos, le transacción debe ser errónea y se debe mostrar la pantalla adecuada.
- Valor de la transacción superior al cupo de la cuenta.
- El módulo 10 de la tarjeta es erróneo: Si se desliza una tarjeta cuyo parámetro de módulo 10 está activo y al realizarse el chequeo de módulo 10 éste es erróneo, se debe mostrar una pantalla indicándolo.

Estas pruebas se realizaron para saber si el dispositivo funciona correctamente en todos los casos y si muestra todas las pantallas necesarias. Al igual que las pruebas anteriores, estas se realizaron constatando en la página de Internet las tramas enviadas y recibidas por Wi-POS, cifradas y en claro. Los resultados fueron los esperados, demostrando que Wi-POS está enviando las tramas necesarias debidamente cifradas, que está recibiendo, descifrando y almacenando las tramas entrantes, y que además de ello los otros procesos como verificación de fecha de vencimiento, creación y cifrado de la trama de PINBLOCK y lectura de la tarjeta de banda magnética funcionan correctamente. Además que se muestran avisos gráficos en caso de que ocurra

cualquier error, esto es importante ya que asegura que el equipo no está almacenando información errónea.

5.4 ANÁLISIS DE RESULTADOS.

Después de probar todos los aspectos del dispositivo como se describió en las secciones anteriores se encontraron los siguientes aspectos importantes:

- El módulo de cifrado funciona de manera correcta, ya que en ninguna de las pruebas que involucraron cifrado de datos obtuvieron errores. Esto indica que la conexión de la FPGA con la memoria serial (ver sección 4.2) está bien y que el ruido no afecta las señales que interconectan al módulo con el microcontrolador, cosa que si ocurría en el montaje en protoboard. Este aspecto es de gran importancia porque así se garantiza la seguridad de la transacción, lo cual estaba planteado en los objetivos. Esto además asegura que los mensajes enviados y la información almacenada de los mensajes recibidos sea correcta, ya que un pequeño error en el cifrado o descifrado altera totalmente el mensaje en claro.

- Las rutinas del microcontrolador (ver sección 4.1.6) se realizan de manera satisfactoria. Todas las tramas son armadas de la manera correcta y la información recopilada y almacenada como estaba previsto.

- La aplicación en J2ME corre como estaba esperado mostrando las pantallas necesarias para la realización de la transacción y manteniendo una comunicación con el servidor en donde se envían y reciben las tramas necesarias para llevar a cabo la transacción. Además maneja de manera satisfactoria el dispositivo, es decir mantiene la comunicación indicada con el microcontrolador para que este realice el proceso necesario en el momento requerido.

- El equipo se encarga de construir de manera correcta las tramas que envían la información necesaria para realizar una transacción bancaria haciendo uso del servidor de ASIC, de esta manera empleando el dispositivo se pueden realizar transacciones bancarias. Con lo que no cuenta el dispositivo es con los permisos necesarios para su uso comercial. Lograr esos permisos y hacer pruebas usando el Tandem de VISA requiere de la firma de un acuerdo comercial como proveedor cuyo primer paso es un proceso de certificación.

Al hacer las pruebas se observó que en la realización de las transacciones hay una demora cuando se recibe información del servidor debido a que el equipo móvil Motorola envía muy lentamente la información por su puerto serial. Este retardo no se debe a la tasa de transmisión del equipo(se uso de 9600bps ver sección 4.1.1) sino a la manera en que se programo el manejo del puerto serial en la aplicación realizada para el teléfono. Esto no afecta los objetivos de éste proyecto pero es un tema que se puede mejorar en un futuro haciendo que la transacción se realice en un tiempo considerablemente más corto.

En ciertas ocasiones se presentaron errores de comunicación en el puerto serial del teléfono debido a que ciertos datos son ingresados muy rápidamente y el teléfono no los recibe de manera adecuada, es por esto que se usa un indicador (LED) que le muestra al usuario cuando el equipo esta listo para recibir la información.

5.5 COSTOS DEL PROYECTO

A continuación se presenta una tabla que contiene la información de los costos del proyecto.

TABLA III Costos del proyecto

Recursos Humanos	Horas de trabajo	Valor Hora	Total
Director Proyecto	96	$ 25.000,00	$ 2.400.000,00
Asesores	40	$ 20.000,00	$ 800.000,00
Desarrolladores área electrónica	3000	$ 20.000,00	$ 60.000.000,00
Total Recursos Humanos			$ 63.200.000,00
Equipo	Cantidad	Valor Unidad	Valor Total
Microcontrolador PIC 18F452 montaje superficial	1	$ 20.000,00	$ 20.000,00
Lector Banda Magnética OMRON v3a	1	$ 64.500,00	$ 64.500,00
Integrado MAX-232	1	$ 4.000,00	$ 4.000,00
Integrado 74LS125A	1	$ 2.300,00	$ 2.300,00
FPGA Altera ACEX1K50TI144-2	1	$ 135.700,00	$ 135.700,00
Memoria serial Altera EPC2	1	$ 42.200,00	$ 42.200,00
Reguladores	2	$ 1.000,00	$ 2.000,00
Integrado 74C922	1	$ 25.000,00	$ 25.000,00
Conector DB9	3	$ 800,00	$ 2.400,00
Alquiler Equipo Motorola i58sr y servicio	1	$ 450.000,00	$ 450.000,00
Alquiler Computador con licencias de software	1	$ 1.000.000,00	$ 1.000.000,00
Alquiler de equipo de Laboratorio	1	$ 1.500.000,00	$ 1.500.000,00
Circuito Impreso	1	$ 120.000,00	$ 120.000,00
Diseño y manufactura del empaque	1	$ 350.000,00	$ 350.000,00
Teclado	1	$ 4.500,00	$ 4.500,00

Impresora	1	$ 350.000,00	$ 350.000,00
Otros (Resistores, condensadores, espadines)	1	$ 10.000,00	$ 10.000,00
Total Equipo			$ 4.082.600,00
Papelería	Cantidad	Valor Unidad	Total
Papel	5 resmas	$ 8.000,00	$ 40.000,00
Encuadernación	6	$ 25.000,00	$ 150.000,00
Empaste	1	$ 4.000,00	$ 4.000,00
Cartuchos de impresora	1	$ 45.000,00	$ 45.000,00
CD	2	$ 2.000,00	$ 4.000,00
Total Papelería			$ 243.000,00
Costos Indirectos	Cantidad	Valor Unidad	Total
Energía	840	$ 161,00	$ 135.240,00
Transporte	1728	$ 1.100,00	$ 1.900.800,00
Valor total			$ 67.525.600,00

6. Conclusiones

El trabajo de grado realizado obtuvo como resultado un dispositivo para realización de transacciones bancarias por medio de una red inalámbrica iDEN. Es un dispositivo pequeño, lo que lo hace cómodo de cargar y sencillo de manipular. Su realización involucró desarrollos en hardware y en software y sus objetivos se cumplieron satisfactoriamente. A continuación se presentan una serie de conclusiones surgidas durante la realización del mismo.

- Gracias a la disponibilidad de equipos y herramientas con que se cuentan en el desarrollo de un trabajo de grado, éste abre un espacio en el cual se pueden generar productos innovadores y realizar proyectos complejos que de otro modo serían muy complicados y costosos de llevar a buen término.
- J2ME demostró ser una poderosa herramienta de fácil manejo para la programación de equipos móviles, lo que hace que proyectos como éste motiven y generen nuevas ideas para desarrollos que involucren el uso de aplicaciones de red y de telefonía móvil. Se utilizó el lenguaje J2ME porque es el lenguaje con el cual se programan los equipos móviles Motorola i85s e i58sr, además por ser un lenguaje fácil de implementar y de gran conocimiento a nivel mundial ya que la comunidad de programadores de J2ME es muy amplia. Las licencias de software de desarrollo para J2ME y para los equipos Motorola son gratuitas y la información sobre actualizaciones e implementaciones de diferentes elementos de este lenguaje son de fácil acceso para todo el mundo.
- Las herramientas de desarrollo de ALTERA proporcionan facilidad para la realización de algoritmos complejos implementados en hardware. En el caso del dispositivo desarrollado en el trabajo de grado, que requería del cifrado en DES por hardware, permitió su implementación sin mayores inconvenientes ya que la herramienta de desarrollo cuenta con posibilidades de simulación que permiten estar seguros del funcionamiento del programa antes de cargarlo en la FPGA. Para la implementación de DES se manipularon bloques de 64 bits, que en caso de haberse hecho en el microcontrolador hubiera tomado más

tiempo de desarrollo y hubiera aumentado considerablemente los tiempos de respuesta por la velocidad de procesamiento y por la obligación de manipular datos de 8 bits de longitud.

- La red iDEN de AVANTEL y sus teléfonos móviles Motorola, permiten el desarrollo e implementación de proyectos que involucran hardware y software, abriendo la posibilidad a la realización de trabajos innovadores que involucran comunicaciones a través de las redes inalámbricas. En el caso de Wi-POS su implementación se puede expandir a otros tipos de redes.
- El desarrollo de este trabajo de grado permitió implementar de forma integral diferentes áreas de la electrónica lo cual demostró que los conocimientos adquiridos a lo largo de la carrera permiten contar con las herramientas y criterios necesarios para realizar desarrollos innovadores y útiles. Conceptos de áreas como Transmisión de Datos, Integración de Redes y el manejo de Sistemas Digitales dentro de este proyecto, así como el manejo de diversos lenguajes de programación, permitieron la realización exitosa de este Trabajo de Grado.
- La ingeniería es una carrera que abre las puertas hacia la riqueza, ya que a partir de un producto creado se puede generar trabajo para muchas personas. Esta afirmación no implica el enriquecimiento de pocas personas sino de aquellas que de una u otra manera se ven afectadas positivamente en el desarrollo, elaboración y comercialización de un producto.
- Para el diseño y desarrollo de un producto se necesita de la colaboración de expertos en varias áreas, a veces la autonomía intelectual del ingeniero electrónico conlleva a que los proyectos mueran por falta de fuerza en otras áreas del desarrollo del producto**.**
- La realización del proyecto hizo notar que los protocolos y estándares de comunicación existentes permiten la integración de distintos módulos con relativa facilidad y esto permite que trabajos como este puedan llegar a implementar con diferentes tipos de dispositivos.

7. Bibliografía

- TANENBAUM, Andrew. **Redes de computadores**. México D.F. Prentice Hall. 1997, 753 p.
- KEOGH, James. **J2ME: The Complete Reference**. Berkeley, California: McGraw-Hill, 2002. 745 p.
- BECERRA, Cesar. Los 600 Principales métodos del Java.
- FEDERAL INFORMATION PROCESSING STANDARDS. Publicación 46-3. Data Encryption Standard (DES). Estados Unidos, 1999. 26 p. (FIPS PUB 46-3).
- STALLING, William Cryptography and Security
- http://idenphones.motorola.com/iden, Motorola iDEN.
- http://java.sun.com/j2me, Lenguaje de Programación J2ME de Java.
- http://www.geocities.com/45peter/iden.html, How iDEN Works.
- http://www.iec.csic.es/criptonomicon/articulos/expertos30.html, Seguridad en los Nuevos Medios de Pago.
- http://www.kanecal.net/mag-stripe-reader-scanner.html, Magnetic Readers.
- http://www.magtek.com/documentation/public/99800004-1.pdf, Magnetic Stripe Card Standards.
- http://www.valhallalegends.com/docs/magcards.htm, Lectores de Banda Magnética.
- http://www.altera.com hojas de especificaciones ACEX1k100.
- Patiño Paola Helena, Rubio Martín Alberto. Diseño e implementacion de encriptore desencriptor triple DES(3DES) en FPGA. Trabajo de grado. Universidad Javeriana, Facultad de Ingenieria, Departamento de electrónica.

ANEXO 1
MARCO TEORICO

1.A DES (DATA ENCRYPTION STANDARD):

DES (*Data Encryption Standard*) es un sistema estándar de uso internacional para convertir datos en secuencias de bits, que carecen de significado, cuando se encuentran en tránsito por un medio de comunicación. Esta información entra a un sistema en el que se codifica de forma binaria mediante algoritmos matemáticos y los procedimientos de cifrado son realizados a partir de una llave que se encarga de cifrar/descifrar los datos. Esta llave es de conocimiento exclusivo del sistema que realiza el proceso.

En 1999 la NIST (*National Institute of Standard and Technology*) publicó por medio de la FIPS (*Federal Information Processing Stardards Publications*), la última versión del estándar DES que fue nombrado como el documento FIPS 46-3, en donde se explican los algoritmos matemáticos para el proceso de cifrado y descifrado de información.

Una llave DES tiene una extensión de 64 bits de los cuales 56 bits son utilizados en el proceso de cifrado, los otros 8 pertenecen a la paridad impar de cada bloque de 8 bits. La información sólo es recuperada por medio de la misma llave que fue usada para ser cifrada. Quien posea la información cifrada y conozca el funcionamiento del sistema de cifrado no puede encontrar la información original sin tener conocimiento de la llave de cifrado, a menos que intente determinar la llave por medio de fuerza bruta, es decir, probando todas las llaves posibles.

Introducción al Algoritmo de cifrado DES

Este algoritmo está diseñado para cifrar y descifrar bloques de información de 64 bits bajo el control de una llave de 64 bits. El descifrado debe realizarse con la misma llave con que fue cifrado, pero con un orden diferente de las subllaves generadas explicadas más adelante.

Un bloque para ser cifrado es sometido primero a una permutación inicial IP (Ver tabla 1A) en donde el orden de los bits se altera teniendo en cuenta la tabla, luego a una computación compleja dependiente de la llave ingresada y finalmente a una permutación inversa IP^{-1} (Ver tabla 2A).

La computación realizada sobre la llave es definida en términos de la función f, llamada función de cifrado, además de una función KS, llamada conmutación de llaves.

Cifrado

Un esquema general de proceso de cifrado es visto en la figura 1A.

Los 64 bits de la trama a ser cifrada son primero sometidos a la permutación de la tabla 1A, llamada permutación inicial **IP.**

Donde el bit 58 de entrada es ahora el primer bit, el bit 50 de entrada es el segundo, así sucesivamente hasta que el bit 7 de entrada sea ahora el último. La salida de esta computación es luego la entrada de la computación compleja dependiente de la llave. Este resultado generado será ahora la entrada de la permutación inicial inversa, que se puede ver en la tabla 2A.

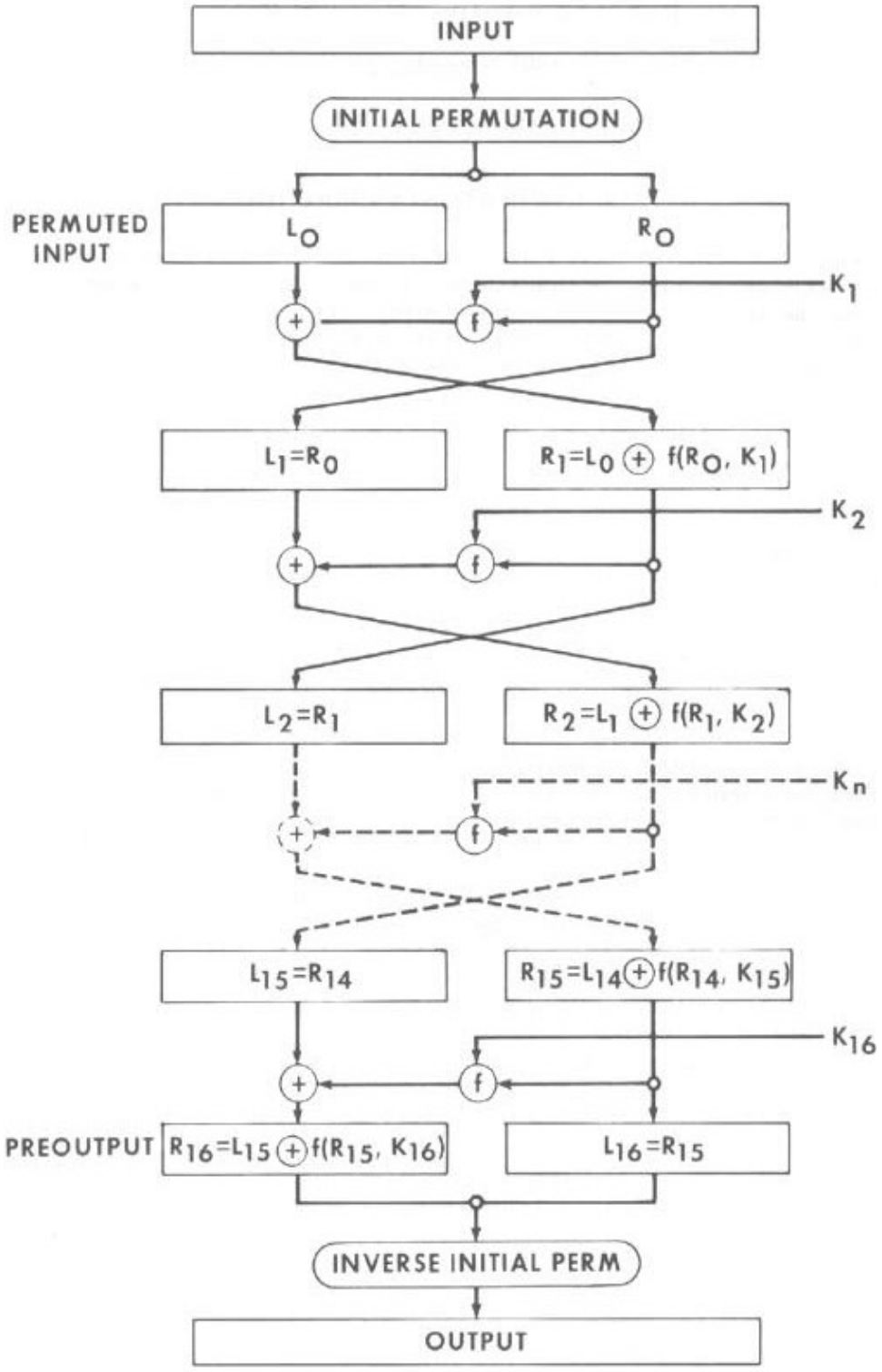


FIGURA 1A. Esquema general del proceso de Cifrado

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Tabla 1A. IP (Permutación Inicial)

IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Tabla 2A. Permutación Inicial Inversa

Donde el bit 40 de la salida del algoritmo complejo dependiente de la llave es ahora el primer bit, el 8 bit el segundo y así hasta que el bit 25 sea el último.

La computación dependiente de la llave consiste en 16 iteraciones de bloques intercambiados en cada paso, explicados mas adelante, en términos de la función f, la cual opera con una entrada de un bloque de 32 bits y otro bloque de 48 bits, teniendo como salida un bloque de 32 bits.

El bloque de entrada de 64 bits proveniente de la permutación inicial es dividido en 2 bloques de 32 bits, uno llamado L y otro llamado R. La entrada a esta computación se puede definir como LR.

K es un bloque de 48 bits generados por la llave de cifrado, que será explicada más adelante. Por lo tanto el bloque L^1R^1 de la primera iteración es definido mediante la

siguiente función.

$$L' = R$$
$$R' = L \oplus f(R,K)$$

Donde $\oplus$ significa la operación XOR.

Para la segunda iteración el bloque de entrada será R^1L^1 y el bloque K será uno diferente al anterior dependiendo de la llave de cifrado.
Con más notaciones podemos describir la iteración con más detalles. Llamemos KS como la función que toma un entero n en el rango de 1 a 16 como entrada además de la llave. Esta función tiene como salida un bloque de 32 bits.

$$K_n = KS(n,KEY)$$

Con K_n determinado por las diferentes posiciones de la llave que serán explicados más adelante.

Ahora llamaremos la salida de la permutación inicial el bloque L_0R_0 de acuerdo al número de la iteración. Por lo tanto en las siguientes funciones tenemos un término más general para todo el proceso de computación compleja.

$$L_n = R_{n-1}$$
$$R_n = L_{n-1} \oplus f(R_{n-1},K_n)$$

Donde 1 toma los valores entre 1 y 16 según sea la iteración. La salida final de este proceso es el bloque $R_{16}L_{16}$.

Descifrado:

El proceso de descifrado es similar a los anteriores salvo en el orden de los bloques K ingresados a la función f. En la siguiente ecuación se muestra el cambio de esta ecuación.

$$R_{n-1} = L_n$$
$$L_{n-1} = R_n \oplus f(L_n, K_n)$$

Donde el orden de K es el inverso al anterior.

Función de cifrado f:

Un esquema del cálculo de la función f(R, K) esta dado en la Figura 2A.

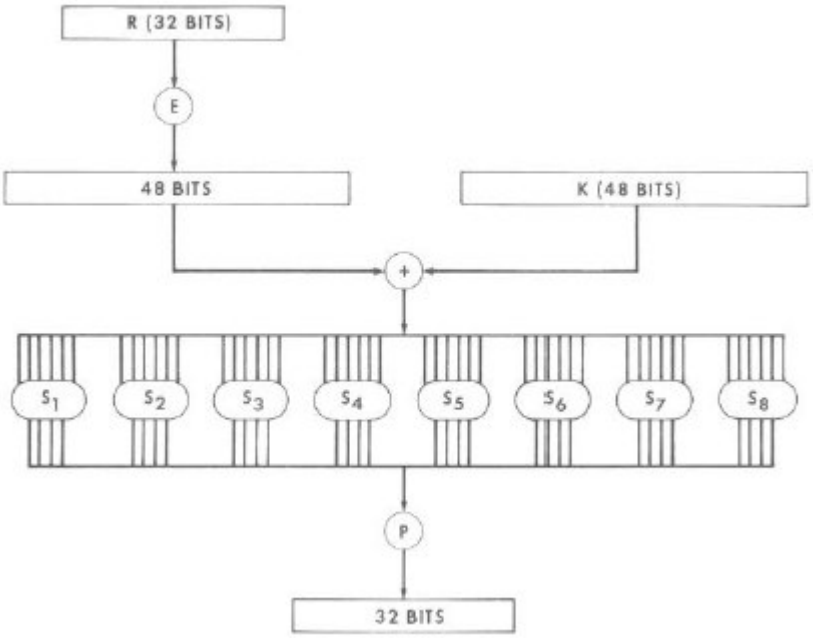


FIGURA 2A, Calculo de f(R, K)

E es una función la cual convierte un bloque de entrada de 32 bits en un bloque de salida de 48 bits, mediante la conmutación de bits descritos en la tabla 3A.

E

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

TABLA 3A. Tabla de expansión E

La salida de la expansión E es operada por medio de una XOR con una subllave de 48

bits que será explicada mas adelante. Este resultado de 48 bits es luego separado en 8 grupos de 6 bits, con los cuales se introducen en las tablas de sustitución de la siguiente manera, como ejemplo se tomará la primera tabla de sustitución ilustrada en la Tabla 4A.

S_1

Row No.	Column Number															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

TABLA 4A. Tabla de sustitución S1

Para la tabla de sustitución S1, se debe usar el primer bloque de 6 bits llamado B, donde los bits serán divididos de la siguiente manera, B_1B_6 representarán la fila de la tabla de sustitución en binario, teniendo como valor mínimo 0 y valor máximo 3, es decir si B_1B_6 es igual a 10, la fila será la número 2, los bits $B_2B_3B_4B_5$ corresponden a la columna de la tabla de sustitución, teniendo como valor mínimo 0 y valor máximo 15. De esta manera se tiene un punto coordenado en la tabla donde se encuentra un valor en binario entre 0 y 15.

Este proceso se realiza con las demás bloques de 6 bits correspondiéndole a cada uno una tabla de sustitución S. Las tablas de sustitución se encuentran en la Tabla 5A y Tabla 6A.

S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

TABLA 5A. Tablas de Sustitución S1, S2, S3 y S4

S_5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

TABLA 6A. Tablas de sustitución S5, S6, S7 y S8

Finalmente se concatenan los resultados de todas las tablas de sustitución en un sólo valor. Como cada tabla de sustitución entrega un valor de 4 bits, el resultado final

tendrá un tamaño de 32 bits. Este valor es sometido a la permutación según se puede ver en la tabla P de la Figura 7A.

P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

TABLA 7A Tabla de permutación P

Generación de las subllaves:

Por cada iteración en la computación compleja se genera una subllave, por lo tanto en total se tienen 16. El cálculo de las 16 subllaves esta descrito en el esquema de la Figura 3A.

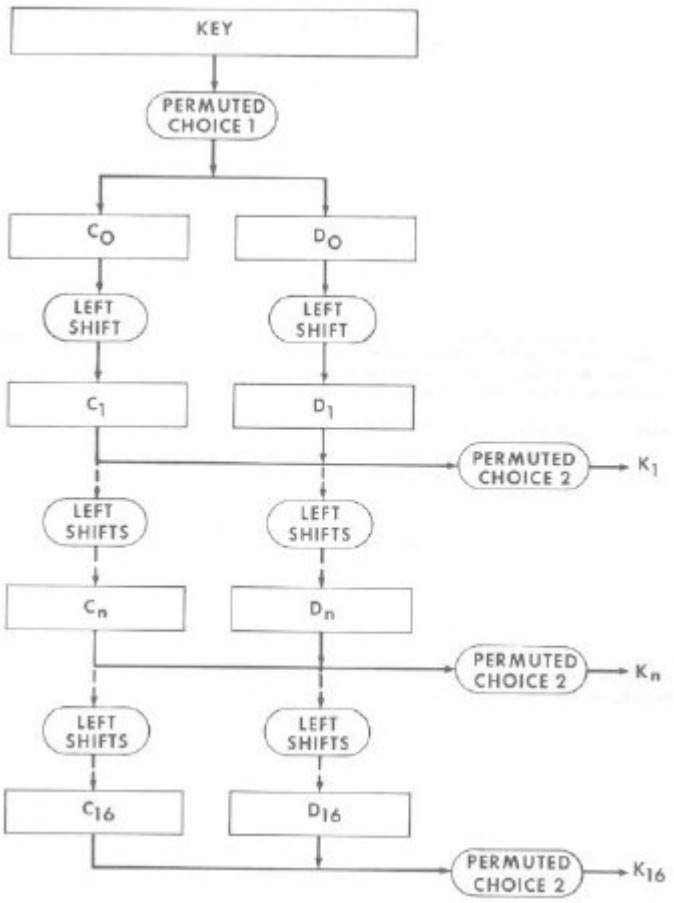


FIGURA 3A . Cálculo de las 16 subllaves

Este proceso consiste en el paso de la información por dos tablas de permutación PC1 y PC2 y por un corrimiento generado según el número de la iteración.

La tabla de permutación PC1 se encuentra descrita en la tabla 8A.

PC-1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

TABLA 8A. Tabla de Permutación PC-1

La tabla ha sido dividida en dos grupos donde se describe como se divide la información en dos bloques de 28 bits donde los primeros toman el nombre de C_0 y los segundos toman el nombre de D_0. Luego, estos bloques son sometidos cada uno a un corrimiento hacia la izquierda según el número de la iteración como lo ilustra la tabla de la Tabla 9A.

Iteration Number	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Tabla 9A. Corrimientos a la izquierda de los bloques C_n y D_n según el número de la iteración.

Finalmente el resultado de la iteración del bloque C_n y D_n es unido formando el bloque C_nD_n que es pasado por la tabla de permutación PC-2 que se caracteriza porque de 56 bits que le entran salen 48 bits.

PC-2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

TABLA 10A. Tabla de permutación PC-2

De esta forma se generan 16 subllaves según la iteración, que serán introducidas en la

función f, explicada anteriormente.

Tarjetas De Banda Magnética[5].

Una tarjeta de banda magnética es un elemento que almacena información mediante el uso de materiales ferromagnéticos, los cuales tienen la capacidad de almacenar datos en forma de campo magnético. Cada tarjeta de banda magnética puede almacenar información en tres pistas distintas. Cada una de estas pistas se usa para guardar cierto tipo de información como se describe a continuación.

- **PISTA 1 (*International Air Transport Asociation* (IATA)):** Posee una densidad de almacenamiento de 210 bits por pulgada. Cada carácter puede ser configurado a partir de 7 bits incluyendo su paridad y puede guardar hasta 79 caracteres alfanuméricos.

- **PISTA 2 (*American Bankers Asociation* (ABA)):** Tiene una capacidad de almacenamiento de 75 bits por pulgada. Cada carácter puede ser representado a partir de 5 bits incluyendo su paridad y tan sólo puede almacenar 40 caracteres alfanuméricos.

- **PISTA 3:** Puede almacenar 210 bits por pulgada. Sus caracteres son descritos a partir de 5 bits incluyendo la paridad y puede almacenar hasta 107 caracteres alfanuméricos.

Estas tarjetas de banda magnética están definidas por el estándar internacional ISO-7811 el cual regula los estándares en cuanto al manejo de la información que transportan las tarjetas que usan este tipo de almacenamiento.

A continuación se muestra como esta distribuida la información en la Pista II.

[5] *Magnetic Readers,* < http://www.kanecal.net/mag-stripe-reader-scanner.html>

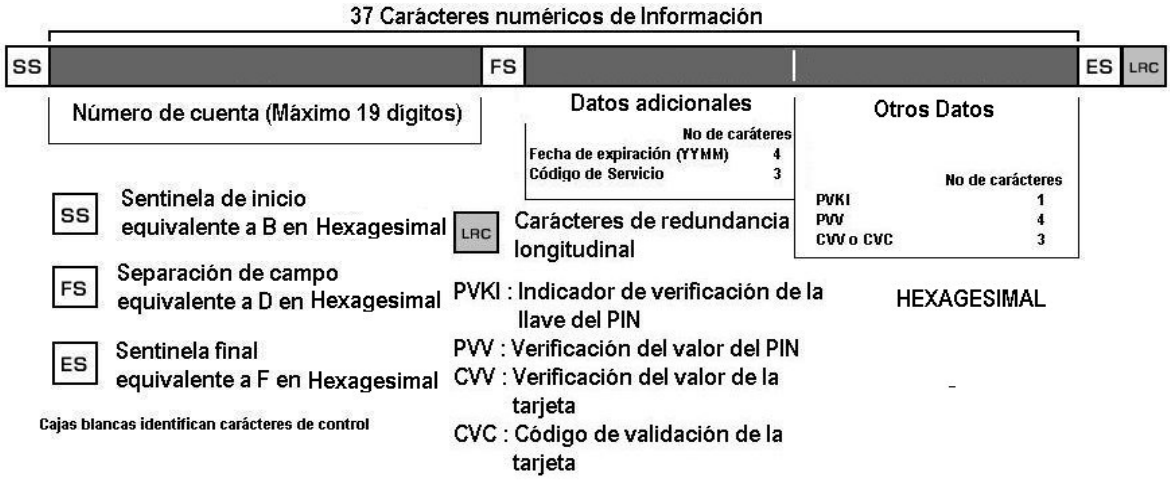


FIGURA 4A. TRAMA DE DATOS DE TARJETAS DE BANDA MAGNÉTICA TIPO PISTA II

Las tarjetas de crédito y débito almacenan información en las Pistas I y II; en la siguiente figura se muestra la distribución de la información en la Pista I.

FIGURA 5A. TRAMA DE DATOS DE TARJETAS DE BANDA MAGNÉTICA TIPO PISTA I

Los lectores de banda magnética consisten en dos solenoides, uno que se encarga de inducir corriente a la tarjeta y otro que se encarga de leer los picos de voltaje que se le inducen a la tarjeta. Al tener esta información, el lector se encarga de amplificar la señal y procesarla para poder manipularla fácilmente.

Magnetic Stripe Card Standards, <http://www.magtek.com/documentation/public/99800004-1.pdf>

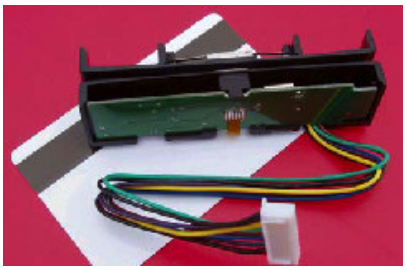


FIGURA 6A LECTOR DE BANDA MAGNETICA

Bloque de PIN (Personal Identification Number):

El bloque de PIN es la forma en que se debe enviar la información del número de cuenta y la clave del usuario cifrada con llave de PIN bajo un estándar. Este bloque tiene una longitud de 64 bits y es usado únicamente en las transacciones bancarias en donde se debe digitar la clave del usuario. El bloque de PIN tambien conocido como PINBLOCK se genera usando una serie de ecuaciones logicas sobre el PIN(clave personal) y el PAN(numero de cuenta) del usuario, este bloque es creado por seguridad .

Módulo 10:

Es un proceso matemático que se utiliza para verificar si el número de una tarjeta débito o crédito es válida haciendo operaciones con sus dígitos. Primero se mira si la cantidad de números de la tarjeta es par o impar, si es par, el primer número de la tarjeta se multiplica por 2, si no, el primer número de la tarjeta se multiplica por 1.

Luego de determinar si es par o impar se multiplican todos los números de la tarjeta alternadamente entre 2 y 1, por lo tanto si es par, se comienza con 2, se sigue con 1, se continúa con 2 y así sucesivamente. Si es impar se comienza con 1 se sigue con 2 y así sucesivamente, si alguno de los dígitos al multiplicarlo da un valor mayor a 9, se debe restarle 9. El proceso se realiza comenzando con el número mas a la derecha.Finalmente se deben sumar todos los valores y el resultado final debe ser un múltiplo de 10 para que el número de la tarjeta sea válido.

Tecnología iDEN.

La red iDEN (*Integrated Digital Enhanced Network*) fue introducida en 1993 por Motorola en Estados Unidos y Japón, y en 1994 mundialmente[6]. Ésta abrió al mercado una nueva generación de soluciones inalámbricas diseñadas para diferentes estilos de aplicaciones móviles enfocadas a los negocios. Hoy en día los equipos inalámbricos iDEN son utilizados en una gran variedad de ambientes de trabajo así como en telefonía móvil comercial.

Los usuarios de equipos Motorola iDEN están encontrando nuevas aplicaciones y descubriendo soluciones de comunicaciones únicas para hacer que sus negocios evolucionen y crezcan. Por ejemplo, las soluciones Motorola iDEN ofrecen la posibilidad de mantener una conferencia con un gran número de personas con sólo oprimir un botón, eliminando la pérdida de tiempo y el costo de realizar llamadas individuales.

La tecnología iDEN permite a sus usuarios aprovechar las ventajas de las aplicaciones inalámbricas avanzadas con un equipo móvil digital que combina: radio digital de dos vías, teléfono inalámbrico digital, mensajes alfanuméricos y transferencia de datos vía Internet. La tecnología iDEN también ofrece un sistema completo de comunicaciones que incluye comandos de voz, agenda telefónica, correo de voz, Internet y e-mail móviles y módems inalámbricos que permiten recrear virtualmente una oficina en cualquier lugar.

La nueva generación de equipos Motorola incorpora la tecnología J2ME ofreciendo la posibilidad de ejecutar aplicaciones interactivas, desde poderosas herramientas de negocios hasta juegos con alto contenido gráfico.

---

[6] *Motorola iDEN,* < http://idenphones.motorola.com/iden/application?namespace=main>

Transacciones Financieras

Las transacciones financieras son procesadas por redes de Intercambio Financiero compuestas por computadores que en conjunto facilitan la transferencia de fondos en línea. Existen cinco componentes principales que participan en una transacción financiera: Tarjetahabiente, Establecimiento Adquiriente, Switch Bancario, Emisor de Tarjetas y Autorizador de Transacciones, entre los cuales cuatro son los que conforman las redes de Intercambio Financiero: Establecimiento Adquiriente, Switch Bancario, Emisor de Tarjetas y Autorizador de Transacciones.

- **Tarjetahabiente:** Es la persona propietaria de la tarjeta débito o crédito
- **Establecimiento Adquiriente**: Esta representado por uno o más computadores que están conectados a los cajeros automáticos, Cajas registradoras, dispositivos de Internet Voice Response (IVR) o PC de Comercio Electrónico por Internet de los diferentes establecimientos comerciales, los cuales introducen transacciones financieras a la red.
- **Switch Bancario**: Esta representado por uno o más computadores que dirigen las transacciones de múltiples adquirientes al respectivo emisor de tarjetas.
- **Emisor de Tarjetas**: Es la institución financiera que tiene una cuenta relacionada con el consumidor.
- **Autorizador de Transacciones**: Maneja la validación y autorización de transacciones contra los parámetros definidos por el Emisor de Tarjetas. Puede estar ubicado en el mismo Emisor de tarjetas, el switch bancario o en una Entidad Independiente.

La siguiente figura describe el escenario común del procesamiento de transacciones financieras, con los componentes nombrados anteriormente:

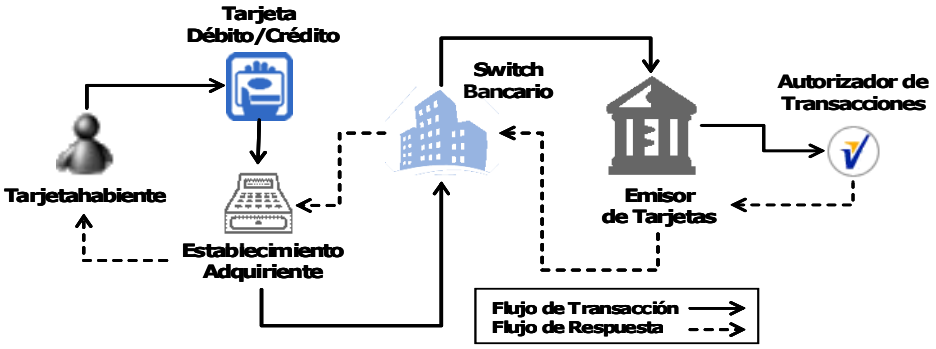


FIGURA 7A. COMPONENTES DE UNA TRANSACCIÓN BANCARIA

Mensajería ISO8583-BASE24

BASE24, es un lenguaje de comunicación interna que manejan los autorizados de transacciones bancarias. El Administrador de Mensajería ISO 8583/BASE24 maneja el formato de los mensajes y requerimientos a nivel de protocolo de transacciones enviadas y recibidas hacia y desde un servidor del Emisor de Tarjetas, basándose en los archivos de configuración del mismo. Adicionalmente maneja las funciones de almacenamiento y envío de transacciones; y ejecuta requerimientos criptográficos como cifrado/descifrado de PIN y cambio dinámico de llaves de cifrado de PIN.

El Mensaje Externo BASE24 está basado en el mensaje externo estándar desarrollado por la Organización Internacional de Estandarización ISO. Este es un mensaje de longitud y contenido variable que puede ser configurado de formas diferentes, basándose en el tipo de mensaje que está siendo enviado. Los procesos del Administrador de Mensajería ISO 8583/BASE24 son los procesos responsables de traducir Mensajes Externos BASE24 entrantes y salientes, hacia y desde formatos de mensajes internos de productos específicos BASE24. Los procesos del servidor Interfaz BASE24 crean e interpretan mensajes externos de acuerdo a las especificaciones. El Mensaje Externo BASE24 permite a los mensajes entrantes y salientes ser configurados individualmente por un servidor, dependiendo de la información del mensaje que el servidor desea enviar y recibir.

FIGURA 8A . ADMINISTRADOR DE MENSAJERÍA ISO8385/BASE24

ANEXO 2
RESULTADOS DE ALGUNAS DE LAS PRUEBAS REALIZADAS CON WI-P.O.S.

Prueba de Funcionamiento de cifrado/descifrado en DES

A continuación se muestran algunos de los resultados obtenidos de el cifrado en DES con WI-P.O.S.

Primera prueba (Todos los datos en formato hexadecimal)

Dato (en hexadecimal): ABC125463BCAFBD0
Llave de cifrado: AB345623DBCDE129
Dato cifrado usando el módulo: 34DA135BC205154D
Dato cifrado usando la tarjeta criptográfica: 34DA135BC205154D. Coincide.

Segunda prueba (cifrado)

Dato: 15975ACDE455ADE2
Llave: 0123456789ABCDEF
Dato cifrado módulo: 1A165D46E7AA37A5
Dato cifrado tarjeta: 1A165D46E7AA37A5. Coincide.

Tercera prueba (descifrado)

Dato: 987DEF3DEF145ABF
Llave: 0133FF569932AB54
Dato descifrado módulo: 9378BC9A8722620D
Dato descifrado tarjeta: 9378BC9A8722620D. Coincide.

Cuarta prueba (descifrado)

Dato: 62547AB34F5A65E8
Llave: 987654321045ABDE
Dato descifrado módulo: 1AC298D759C62FC8
Dato descifrado tarjeta: 1AC298D759C62FC8. Coincide.

Pruebas de Funcionamiento de cifrado en DES en modo CBC

A continuación se muestran los resultados de algunas de las pruebas realizadas con WI-P.O.S. al cifrar y descifrar ciertas tramas, cabe anotar que se realizaron muchas mas pruebas.

Primera Prueba (cifrado)

Dato (en hexadecimal): ABC125463BCAFBD0CE45BCADAC5DE435
Llave de cifrado: AB345623DBCDE129
Vector de inicio: 9378BC9A8722620D
Dato cifrado usando el módulo: 69881742C5DC217680FCD24EB9D963BD
Dato cifrado usando la tarjeta criptográfica: 69881742C5DC217680FCD24EB9D963BD. Coincide.

Segunda Prueba (cifrado)

Dato (en hexadecimal): 12345AEFD1D2E2FA89ED2A1B1A2E3E5A2D2E1A1D24AB1234
Llave de cifrado: 12345AB56E3E2E1F
Vector de inicio: E7D845D4E67C6EFD
Dato cifrado usando el módulo: : 80F6253A35E6075864AADB9A3716BC9F3F259789BE0600C2
Dato cifrado usando la tarjeta criptográfica:
80F6253A35E6075864AADB9A3716BC9F3F259789BE0600C2. Coincide.

Tercera Prueba (descifrado)

Dato (en hexadecimal): CADE123E45E67EAFE781231256875423
Llave de cifrado: 1AB3CDEF56ED4DE3
Vector de inicio: E5DC76456EDC4565
Dato cifrado usando el módulo: 043C8A19F711D522121C9F432B4A5E74
Dato cifrado usando la tarjeta criptográfica:
043C8A19F711D522121C9F432B4A5E74. Coincide.

Cuarta Prueba (descifrado)

Dato (en hexadecimal):ABC125463BCAFBD05ABCD3547CDFEC4DE7D845D4E67C6EFDEA
Llave de cifrado: 12DEFEEE54541BCD
Vector de inicio: 1AB3CDEF56ED4DE3
Dato cifrado usando el módulo: 9C0157C8F82030E76F397E30C8EF228544626EC1C8053C9F
Dato cifrado usando la tarjeta criptográfica:
9C0157C8F82030E76F397E30C8EF228544626EC1C8053C9F. Coincide.

Prueba de funcionamiento de carga de llave de transporte y verificación por dígitos de chequeo

A continuación se muestran las pruebas realizadas al cargar diferentes llaves de transporte en WI-POS, por medio de la aplicación realizada en Visual Basic y luego verificadas con la opción de dígitos de chequeo.

Primera Prueba:

Llave de cifrado: 0123456789ABCDEF
Dato cifrado usando tarjeta criptográfica: D5D44FF720683D0D
Dígitos de chequeo obtenidos de la herramienta en Visual Basic: D5D4. Coincide.

Segunda Prueba:

Llave de cifrado: FEDCBA9876543210
Dato cifrado usando tarjeta criptográfica: A68CDCA90C9021F9
Dígitos de chequeo obtenidos de la herramienta en Visual Basic: A68C. Coincide.

Tercera Prueba:

Llave de cifrado: 9876543210987654
Dato cifrado usando tarjeta criptográfica: DF998E5C01B982C9
Dígitos de chequeo obtenidos de la herramienta en Visual Basic: DF99. Coincide.

Prueba de funcionamiento de carga de llave maestra y verificación por dígitos de chequeo

Las siguientes pruebas describen los resultados del ingreso de las llaves maestras por medio de la aplicación en Visual Basic y su verificación con los dígitos de chequeo.

Prueba con una sola llave maestra:

Número de llaves maestras introducidas: 1
Llave maestra 1: 1237894561237894
XOR de Llave maestra: 1237894561237894
Dato cifrado usando tarjeta criptográfica: F027F940D5A45D93
Dígitos de chequeo obtenidos de la herramienta en Visual Basic: F027. Coincide.

Prueba con dos llaves maestras:

Número de llaves maestras introducidas: 2
Llave maestra 1: ACDEF65432FE7891
Llave maestra 2: 001599884411223312
XOR de Llave maestra: B9477E1023DC4B83

Dato cifrado usando tarjeta criptográfica: E63A3D454FD9CDE0
Dígitos de chequeo obtenidos de la herramienta en Visual Basic: E63A. Coincide.

Prueba con tres llaves maestras:

Número de llaves maestras introducidas: 3
Llave maestra 1: A1D5E6B478CCBB0
Llave maestra 2: F896321475FACDEF
Llave maestra 3: 558899665544212358
XOR de Llave maestra: A703F5196732277C
Dato cifrado usando tarjeta criptográfica: 23C13EA287C14ECA
Dígitos de chequeo obtenidos de la herramienta en Visual Basic: 23C1. Coincide.

Pruebas de selección de llave maestra

A continuación se ilustran los resultados obtenidos al seleccionar la llave maestra, verificando con los dígitos de chequeo.

Primera prueba de selección de llave maestra

Llave maestra anterior: 1237894561237894
Dígitos de chequeo obtenidos de la herramienta en Visual Basic: F027
Llave maestra nueva 1: ACDEF65432FE7891
Llave maestra nueva 2: 001599884411223312
XOR de Llave maestra nueva: B9477E1023DC4B83
Se actualiza la llave maestra.
Dígitos de chequeo obtenidos de la herramienta en Visual Basic: E63A

Se selecciona de nuevo la llave maestra anterior y se solicitan los dígitos de chequeo: F027

Segunda prueba de selección de llave maestra

Llave maestra anterior: B9477E1023DC4B83
Dígitos de chequeo obtenidos de la herramienta en Visual Basic: E63A
Llave maestra 1: A1D5E6B478CCBB0
Llave maestra 2: F896321475FACDEF
Llave maestra 3: 558899665544212358
XOR de Llave maestra: A703F5196732277C
Se actualiza la llave maestra.
Dígitos de chequeo obtenidos de la herramienta en Visual Basic: 23C1

Se selecciona de nuevo la llave maestra anterior y se solicitan los dígitos de chequeo: E63A. Con esto se comprobó que tanto la aplicación realizada en Visual Basic como el microcontrolador funcionan correctamente en lo que respecta a selección de llave maestra.

Pruebas de funcionamiento de carga de parámetros iniciales

Para cada una de estas pruebas se usó una llave de transporte diferente y un vector de inicio diferente (cargados en Wi_POS y en el "Banco Virtual"). A continuación se observan los números de terminal consignados en la base de datos y sus parámetros correspondientes.

Tabla 1B. Información asociada a los números de los terminales.

	Número de terminal	IVA	Propina	Dígitos de Clave	Base devolución de IVA	Emisor
1	12345678	16	Habilitado	4	Habilitado	VISACENTRO
2	87654321	12	Deshabilitado	4	Habilitado	UJaveriana
3	13482579	15	Habilitado	5	Deshabilitado	CENTROVISA

Los campos de tipo de mensaje han sido alterados ya que esta información es confidencial. Todas las tramas se encuentran explicadas en detalle en la sección 3.3 para entender los campos que las componen remitirse a esa sección. Las tramas se rellenan con el carácter F para que sean múltiplo de 64 bits y puedan ser cifradas.

i) Prueba con el número de terminal 12345678

Trama cifrada enviada desde Wi-POS al Banco Virtual.(HEXA)	D2041EA29A2BD60E50D3BFBB866830F1
Trama descifrada enviada desde Wi-POS al Banco Virtual(ASCII)	0212345678FFFFFF
Trama de Respuesta de parámetros Iniciales en claro.	031234567816S041VISACENTROFFFFFF
Trama de parámetros de Respuesta Cifrada	EF2FD3A6FE9952488CA3141B4537B500B8C B431C18CE929822428614 69B0E0E4

El proceso se realizó satisfactoriamente y esto se comprobó pidiendo los parámetros iniciales usando las rutinas del microcontrolador desarrolladas para ello (Ver sección 4.1.5).

El tiempo que se tardó en efectuar el proceso fue de 32.42s. Este tiempo es medido desde el momento en que se ingresa a la aplicación en el teléfono hasta que aparece la pantalla de carga de parámetros exitosa (Fig. 23, Cap4). Para la ver la secuencia de carga de parámetros en la aplicación de J2ME vea la sección (4.3.3.)

ii) Prueba con el número de terminal 87654321

Trama cifrada enviada desde Wi-POS al Banco	39338F0332DC0601AB31D8D21B2748B7

Virtual.(HEXA)	
Trama descifrada enviada desde Wi-POS al Banco Virtual(ASCII)	0287654321FFFFFF
Trama de Respuesta de parámetros Iniciales en claro.	038765432112N040UJaverianaFFFFFF
Trama de parámetros de Respuesta Cifrada	260062455B6F6210DA5A409B71FB32C690A3 4A35E62E3087E94EF4EA96FA9C89

El proceso fue satisfactorio y tardó 30.01s.

iii) Prueba con el número de terminal 13482579

Trama cifrada enviada desde Wi-POS al Banco Virtual.(HEXA)	6794DD3DD744F918FB89AA0686289FFD
Trama descifrada enviada desde Wi-POS al Banco Virtual (ASCII)	0213482579FFFFFF
Trama de Respuesta de parámetros Iniciales en claro.	031348257915S050CENTROVISAFFFFFF
Trama de parámetros de Respuesta Cifrada	C15DC5391F0E8917EA7C6629EO63314899FF 59B1223F20A40C160CC0D066019804

El proceso fue exitoso y se llevó a cabo en 29.01s

iv) Prueba con llaves de transporte diferentes.

La siguiente prueba se realizó con llaves de transporte erróneas, es decir una llave de transporte en el Wi-POS y otra diferente en el servidor con el fin de observar si la aplicación en J2ME y el microcontrolador funcionan bien en caso de error, es decir si despliegan la pantalla correspondiente al caso. La prueba demostró que al descifrar incorrectamente la trama en el servidor, se devolvió como respuesta una trama que el

microcontrolador no entendió, de manera que en pantalla apareció que la trama recibida era errónea, como era de esperarse.

v) Prueba con número de terminal no registrado.

Se realizó una prueba más para verificar si se despliega la pantalla correspondiente a un error en los mensajes. Como el número de terminal no estaba registrado se devolvió una trama errónea, de manera que apareció una pantalla indicando lo sucedido, tal como era esperado.

Pruebas de funcionamiento de transacciones

Los parámetros para los usuarios ingresados en la base de datos fueron los siguientes:

Tabla 2B. Usuarios ingresados en la base de datos y sus correspondientes parámetros.

	Usuario	Fecha Vencimiento 1	PIN de Crédito	Solicitud de Cuenta	Valida Fecha de Vencimiento	Voucher	Módulo 10	Solicitud de cuotas	PIN débito
1	Usuario 1	0	0	1	0	1	0	0	1
2	Usuario 2	1	0	0	1	1	1	1	0
3	Usuario 3	0	0	0	1	1	0	0	1
4	Usuario 4	0	0	0	0	0	0	0	1
5	Usuario 5	1	1	1	1	1	1	1	0
6	Usuario 6	0	0	0	0	0	0	0	0

Tabla 3B. Parámetros de cuenta de los usuarios

	Usuario	Tipo de Cuenta	PIN	Cupo	Estado
1	Usuario 1	1	1111	100.000	ACTIVO
2	Usuario 2	2	9988	10.000.000	ACTIVO
3	Usuario 3	0	1234	1.000	ACTIVO
4	Usuario 4	0	2222	85.000.000	ACTIVO
5	Usuario 5	2	9137	78.000.000	ACTIVO
6	Usuario 6	1	6482	21.470.000	ACTIVO

Como el proceso de transacción también depende de los parámetros iniciales, las pruebas se hicieron con distintos números de terminal. A continuación se muestran unas pocas pruebas.

i). Prueba de transacción satisfactoria, con el usuario 1, con el terminal 1.

Para que el proceso fuera satisfactorio se tuvo cuidado de no excederse en el valor y de ingresar correctamente la clave. Se realizó una transacción por 12.000 pesos y se dio una propina de 500 pesos, se digitó correctamente la clave y como tramas enviadas y recibidas al banco virtual se tuvieron las tramas mostradas en la tabla III

A continuación se muestra la trama enviada con la solicitud de parámetros de tarjeta (Para ver los campos específicos de esta trama remitirse a la sección 1.3), donde por cuestiones de seguridad se ha cambiado el número de la tarjeta usada. Nótese que en esta trama se solicitan los parámetros específicos de la tarjeta y como respuesta se reciben los valores consignados en la Tabla III para el "Usuario 1", indicando que se debe mostrar el ingreso de tipo de cuenta, se debe pedir el PIN del usuario y se debe imprimir el comprobante.

Trama cifrada enviada desde Wi-POS al Banco Virtual.(HEXA)	AC210AAABC5E7188305F8AA0A6FC68CCB33350 60F2F9621E5F83D4C9242E68DD
Trama descifrada enviada desde Wi-POS al Banco Virtual (ASCII)	041234567858951825225123226 69FFF
Trama de Respuesta de parámetros de tarjeta en claro.	05123456785895182522512322669001010 01FFF
Trama de parámetros de Tarjeta de Respuesta Cifrada	D0199580CD60C78056F086E563D9FAEF2DA90F8 FC2344880 067C0A5B5FCA16C05BE0187E9FD94708

Debido a que el parámetro de PIN está habilitado, se debe solicitar una llave de cifrado de PIN al Banco Virtual, por lo que se construye la trama de solicitud de llave de PIN (Esta trama se explica con detalles en la sección 3.3), a continuación se muestran los resultados de esta solicitud.

Trama cifrada enviada desde Wi-POS al Banco Virtual. (HEXA)	759ADD16843D83E799E2D6E8F36A8CC8
Trama descifrada enviada desde Wi-POS al Banco Virtual(ASCII)	0012345678FFFFFF
Trama de Respuesta de solicitud de llave de PIN en claro.	01123456788177917481779174FFFFFF
Trama de solicitud de llave de PIN de Respuesta Cifrada	2CE5D8F04645DF1E649537BD0678591BCF657E41 A489E29C677C1002F0C48194

Después de esto Wi-POS pidió a través de la pantalla del teléfono los campos correctos y fueron ingresados de manera correcta. Con la información ya recopilada se armó la trama de solicitud de transacción de pago (esta trama se encuentra descrita en la sección 3.3), a continuación se muestran los resultados.

Trama cifrada enviada desde Wi-POS al Banco Virtual.(HEXA)	E36D75B0ADDBDCDDC9C8E95A37D023A6E2F3A4611E8759BE600C556AC137AF9CDBD4423095AD118E8A8799316E9E048258D33ECEA8495207E51F4000F8636735CE339D7A012D1764A7E8A80B0E440FBA91DF7529997E55135CDA91810355F6A4AA0CCD9B0CBD98E9639F5B6AE520B787DD30C46324FC491A17FD28D0563BB9BDE7686A27A5E51AA7535DFAED2DE8E4A4BB8AE495C2AA0EED4090AB78D920008C1F28B20EFA41231C2FABCEB7DFC8A4FCD4E98B764FA83A44
Trama descifrada enviada desde Wi-POS al Banco Virtual(ASCII)	06123456780000005895181255212322669=491212000023????B589518000001232266?????????????????????????????4912120?????????????????????2C21C0841756E83C100000001300900000000172414000001077588?
Trama de Respuesta de transacción en claro.	071234567800000000transaccion exitosaF327197FFFF
Trama de transacción de Respuesta Cifrada	03D60014DCE740D5AEBC41BF34126E19BDA7B8FEFA6A97EF1A3E01B5A78B58D7913A132BB348A2CB00D83E2E2D46EFE9

La transacción se llevó a cabo satisfactoriamente mostrando la información correcta en el equipo Motorola. Se puede ver que los mensajes enviados fueron los correctos y que se transmitieron debidamente cifrados Al mismo tiempo en el banco virtual se realizó el descuento de la cuenta. La transacción fue hecha por un valor de 12.000 pesos, en la cuenta se tenía un cupo de 100.000, se entregó una propina por 500 pesos, quedando como cupo final un valor de 87.500 pesos.

ii). Prueba de transacción satisfactoria, con el usuario 2, con el terminal 2.

A continuación se muestra una transacción con una tarjeta de crédito, en donde no se

pide al usuario que ingrese el PIN, por lo tanto la solicitud de llave de PIN no es enviada al Banco Virtual.

En la siguiente tabla se muestran las tramas de solicitud de parámetros de tarjeta

Trama cifrada enviada desde Wi-POS al Banco Virtual. (HEXA)	4EC4DFF969AC8EC154F802A22D3E6A73C56DE74236F60AC83E48F298D4DB3ECD
Trama descifrada enviada desde Wi-POS al Banco Virtual (ASCII)	048765432137781340700000=080FFF
Trama de Respuesta de parámetros de tarjeta en claro.	058765432137781340700000=08010011110FFF
Trama de parámetros de tarjeta de Respuesta Cifrada	7B0B52C40C96094903CABBA8D1A440BEBCFF0012F9E02943FA509058DB2D515EC863628DF894E055

Después de esto se envía la solicitud de transacción que se muestra a continuación:

Trama cifrada enviada desde Wi-POS al Banco Virtual.(HEXA)	650237CEF4DA7C39D4822BA2F8FBEA82B49DBEB26A9FB8706A54AD4EDCD707C329DE61D28C9CAB4583D74E27580F821C90913706FDB6D780B95931E87FCC26E9C050B60565D4F6DCFDB79B9F598BA5EB37371B59B1F24518F17E06B50313E0080B52BA7916FF9C330DFF0EB98FF059379E2B5CCA27B03F7CDA7FDCF2D0EB84705BD4CF4C98F46C87EEFE5DC778736182D1754DB75F92AADAF5779432F3F35C7B71AFC2CAAF41B93B114314CFDAE665C17313416475D1035D
Trama descifrada enviada desde Wi-POS al Banco Virtual(ASCII)	06876543210000043778134071111 11=08021010502573780000?B377813407904367?VILLARREAL?NELSON??????????0802101050257378?B????????????0000000000000000

	00700000035700000000003825000000000000?
Trama de Respuesta de transacción en claro.	078765432100000400transaccion exitosaF498199FFFF
Trama de transacción de Respuesta Cifrada	A0B32E7A45325A6E1C84202554A670BE8D2B2C2F779 F4C48F74C13B7225B95A4410A3B1948E2593A813C1A1 8AE180EE9

Los resultados de la transacción fueron satisfactorios y se confirmaron al observar en la tabla de base de datos el descuento al cupo del usuario. Y al analizar las tramas enviadas y recibidas y concluir que eran correctas.

iii). Prueba de transacción satisfactoria, con el usuario 3, con el terminal 3.

Esta prueba de transacción fue realizada con una tarjeta de crédito a la que se le pedía PIN y todos los demás parámetros. A diferencia de los demás terminales, este pide como PIN un número de 5 dígitos.

Primero se pide la trama de solicitud de parámetros de tarjeta, que se muestra en la siguiente tabla.

Trama cifrada enviada desde Wi-POS al Banco Virtual.(HEXA)	B278CFD2A54F396B020C40CD2DAB3AF40008D5 1D9A06942A7B7808EEFD811D88
Trama descifrada enviada desde Wi-POS al Banco Virtual(ASCII)	04134825795183610187611111=12FFF
Trama de Respuesta de parámetros de tarjeta en claro.	05134825795183610187611111=1211111110FFF
Trama de parámetros de Tarjeta de Respuesta Cifrada	7D8091EB13E63C9BEF5E113EBBDDEC907DD131 86BF4C6CC6D7C2DB8062157EA5FE056AEFC73F 1B13

Debido a que entre sus parámetros de tarjeta estaba explícito la solicitud del PIN, se realiza a continuación una solicitud de PIN.

Trama cifrada enviada desde Wi-POS al Banco Virtual.(HEXA)	F9320E1759C664ACAF1B201096D880CC
Trama descifrada enviada desde Wi-POS al Banco Virtual (ASCII)	0013482579FFFFFF
Trama de Respuesta de solicitud de llave de PIN en claro.	011348257932359924323599 24FFFFFF
Trama de solicitud de llave de PIN de Respuesta Cifrada	DF6C17887BFEA7C78911208AFC6F885F5DE38C2 08BC8DD56B23310C489FE7F6E

Después de ingresar los valores solicitados se envía la trama de solicitud de transacción bancaria.

Trama cifrada enviada desde Wi-POS al Banco Virtual.(HEXA)	F6FAA6B9C4C80B81FC38E1BAEE4293974CCD6CB531E239 FD039E8B1442A480192127D349042E9E02DA873DF0497E02 9D96CBA7B54831E990905F28B58F411A387E87075CA35493 A62F58A2169EA87964C0519560EF76BCC74B060B91F8EEF AF9B132B7B59E5F4788002B32AA8AC673057148DC4A3A2D 5EE0AA8B007784B2A5E2C39DE6DA838DD24A928000BB8C 84B2C71DAB1011F5EFC2E6542ED5659F9B68DA1BE9363007 DB78EF4B8D5A91E61A6D0215CE0F64D35E0ABB
Trama descifrada enviada desde Wi-POS al Banco Virtual(ASCII)	06134825790000095183610187611111=12051010000002000000 ?B5183610187616762?PINILLA?GARCIA?DANIEL?F????1205 10100000020?K????????????79333A4780E9E1E3202000001254 70000000016365700000000000?
Trama de Respuesta de transacción en	071348257900000900transaccion exitosaF674942FFFF

claro.	
Trama de transacción de Respuesta Cifrada	8A6B0A2285B744273755E11A21F6BC7ECC93928B3A0F384E BF24790942F96C41F8F244D4CFE3BF8FAF1F541C33A34AA7

La transacción se llevó a cabo de manera correcta y las pantallas mostradas por el teléfono fueron las adecuadas.

De manera similar se hicieron otras pruebas satisfactorias mostrando que el equipo funciona de acuerdo a lo requerido. En algunas de estas pruebas se presentaron errores de comunicación por el puerto serial del teléfono, mostrándose así, la pantalla apropiada para cada caso (Ver sección 4.3).