



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

IMPLEMENTACIÓN DEL PROTOTIPO DE UN SISTEMA DE COBRO CON
DINERO ELECTRÓNICO PARA EL PARQUEADERO DE LA
UNIVERSIDAD DE LAS AMÉRICAS

AUTOR

Jefferson David Guachamin Chamorro

AÑO

2018



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

IMPLEMENTACIÓN DEL PROTOTIPO DE UN SISTEMA DE COBRO CON
DINERO ELECTRÓNICO PARA EL PARQUEADERO DE LA UNIVERSIDAD DE
LAS AMÉRICAS

Trabajo de Titulación presentado en conformidad con los requisitos establecidos
para optar por el título de Ingeniero en redes y telecomunicaciones.

Profesor Guía

Mgtr. Milton Neptalí Román Cañizares

Autor

Jefferson David Guachamin Chamorro

Año

2018

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido el trabajo, Implementación del prototipo de un sistema de cobro con dinero electrónico para el parqueadero de las UNIVERSIDAD DE LAS AMÉRICAS, a través de reuniones periódicas con el estudiante Jefferson David Guachamin Chamorro, en el semestre 2018-1, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.”

Milton Neptalí Román Cañizares

Magister en gerencia de Redes y Telecomunicaciones.

CI: 0502163447

DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber revisado este trabajo Implementación del prototipo de un sistema de cobro con dinero electrónico para el parqueadero de las UNIVERSIDAD DE LAS AMÉRICAS, de Jefferson David Guachamin Chamorro en el semestre 2018-1, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.”

Héctor Fernando Chinchero Villacís

Master en domótica.

CI: 1715451330

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

Jefferson David Guachamin Chamorro

CI: 1725867921

AGRADECIMIENTOS

Agradezco a Dios por el regalo de la vida y por la bendición de tener una familia que siempre estará en todo momento.

A mis padres por el apoyo que me han brindado en mis distintas etapas estudiantiles.

A la Universidad de Las Américas; a todos los ingenieros que me han ayudado a crecer en la vida profesional no solo con conocimientos teóricos sino también valores, especialmente al Ing. Milton Román Cañizares por haberme tomado en cuenta para dirigir esta tesis, apoyándome cuando lo necesité.

DEDICATORIA

A mi madre que con su apoyo he podido obtener el título universitario y a mi abuelo Ismael Gustavo Chamorro Vergara quien fue pionero en las telecomunicaciones en el Ecuador.

Jefferson D. Guachamin Ch.

RESUMEN

En el presente proyecto se desarrolla un prototipo de cobro para parqueadero con dinero electrónico. El sistema está basado en software y hardware libre, los cuales nos permiten interactuar con distintas plataformas como la red GSM y la plataforma de dinero electrónico del Banco Central del Ecuador.

Para empezar se muestran los conceptos sobre el funcionamiento de dinero electrónico en el Ecuador, nociones fundamentales de redes seguido de una introducción a la red GSM, adicionalmente se describe la tecnología USSD y NFC que serán fundamentales en el cobro con dinero electrónico y finalmente se describen tecnologías y herramientas necesarias para la elaboración del prototipo.

En el capítulo dos, se realiza el diseño y se describe la tecnología que se utilizara en los subsistemas que componen el prototipo como, el subsistema de cobro, subsistema de toma y procesamiento de datos y finalmente el subsistema de visualización de data, así como los flujos que deben cumplir.

En el capítulo tres, se presenta la implementación y configuración de los subsistemas que componen el sistema de cobro con dinero electrónico. Adicionalmente se presentan las pruebas realizadas para validar el funcionamiento del prototipo.

Finalmente, en el capítulo cuatro, se presentan las conclusiones adquiridas mediante el desarrollo del proyecto, así como se incluyen recomendaciones para su funcionamiento y mejoras para futuros proyectos que se relacionen con el tema.

ABSTRACT

The following project develops a prototype for charging the parking with electronic money. The system is based in free software and hardware, which allows to interact with different platforms such as GSM networks and the electronic money platform of the Ecuadorian Central Bank.

To start, the project shows the concepts about the functionality of the electronic money in Ecuador, fundamentals of networking and the introduction to a GSM network. In addition, it describes the USSD and NFC technology that are the bases for charging the electronic money; finally, the project describes the necessary technologies and tools to develop the prototype.

In the second chapter, it describes the design and technology that will be used in the subsystems that belongs to the prototype: the charging subsystem, getting and processing subsystem, and the visualization subsystem with the corresponding flows that they have to fulfill for each of them.

In the third chapter, it presents the implementation and configuration of the subsystems and the components of the charging system with electronic money. Furthermore, it indicates the test performed to ensure the correct functionality of the prototype that are considered on the conclusions.

Finally, in the fourth chapter, it presents the conclusions obtained in the project development and the recommendations in the operation and improvements for future projects related to this topic.

ÍNDICE

| | | |
|--------|---|----|
| 1. | FUNDAMENTO TEÓRICO..... | 1 |
| 1.1 | Dinero Electrónico En El Ecuador..... | 1 |
| 1.1.1 | Antecedente | 1 |
| 1.1.2 | Definición De Dinero Electrónico | 2 |
| 1.1.3 | Principales Elementos En El Sistema De Dinero Electrónico..... | 3 |
| 1.2 | Tipos de interfaces tecnológicas que interactúan con dinero electrónico del BCE..... | 4 |
| 1.3 | Nociones Fundamentales De Redes..... | 5 |
| 1.3.1 | La informática centralizada..... | 5 |
| 1.3.2 | La informática descentralizada | 6 |
| 1.3.3 | Redes heterogéneas | 7 |
| 1.3.4 | Redes informáticas actuales..... | 7 |
| 1.4 | Red GSM..... | 8 |
| 1.4.1 | Generalidades De GSM..... | 8 |
| 1.4.2 | Arquitectura De Una Red GSM..... | 8 |
| 1.4.3 | Centro De Autenticación..... | 9 |
| 1.4.4 | Registro De Identificación De Equipos | 9 |
| 1.4.5 | Registro De Localización De Visitante..... | 10 |
| 1.4.6 | Registro De Ubicación Base..... | 10 |
| 1.4.7 | Centro De Comunicación Móvil | 10 |
| 1.4.8 | Controlador De La Estación Base..... | 11 |
| 1.4.9 | Estación Móvil..... | 11 |
| 1.4.10 | Módulos GSM | 12 |
| 1.5 | Tecnología USSD..... | 14 |
| 1.5.1 | Servicios Prestados De Red..... | 14 |
| 1.5.2 | Servicios Portadores..... | 15 |

| | | |
|---------|--|----|
| 1.5.3 | Teleservicios..... | 15 |
| 1.5.4 | Servicios Suplementarios | 15 |
| 1.5.5 | Funcionamiento Y Características | 16 |
| 1.5.6 | MMI..... | 17 |
| 1.5.7 | Modo Aplicación | 18 |
| 1.5.8 | Códigos USSD Más Conocidos En El Ecuador | 19 |
| 1.6 | Microcomputadores | 20 |
| 1.6.1 | Raspberry PI..... | 20 |
| 1.6.2 | Principales Elementos Del Raspberry PI | 21 |
| 1.6.3 | Módulo NFC/RFID | 23 |
| 1.7 | Tecnologías Y Herramientas | 23 |
| 1.7.1 | Bases De Datos Sql..... | 23 |
| 1.7.1.1 | MaríaDB..... | 24 |
| 1.7.2 | Html | 24 |
| 1.7.3 | Php..... | 24 |
| 1.7.4 | Python | 25 |
| 1.8 | Sistemas Operativos | 25 |
| 1.8.1 | Windows | 25 |
| 1.8.2 | MacOS..... | 26 |
| 1.8.3 | GNU/Linux..... | 26 |
| 1.8.4 | Raspbian | 26 |
| 1.9 | Tecnología NFC..... | 26 |
| 1.9.1 | Definición De La Tecnología NFC | 26 |
| 1.9.2 | Especificaciones Técnicas..... | 27 |
| 1.9.3 | Tipos Y Modos De Comunicación..... | 28 |
| 1.9.4 | Limitaciones NFC | 30 |
| 2. | DISEÑO DEL PROTOTIPO | 30 |
| 2.1 | Requerimientos Del Sistema..... | 30 |
| 2.2 | Funcionamiento Del Prototipo..... | 34 |

| | | |
|---------|---|-----------|
| 2.3 | Diseño De Subsistema De Cobro..... | 34 |
| 2.3.1 | Tecnologías Aplicadas Al Subsistema De Cobro..... | 35 |
| 2.3.1.1 | Módulo GSM..... | 36 |
| 2.3.1.2 | Plataforma De Cobro Con Dinero Electrónico BCE | 36 |
| 2.3.1.3 | Terminales Móviles | 36 |
| 2.4 | Diseño Subsistema De Toma Y Procesamiento De Datos | 36 |
| 2.4.1 | Bloque De Toma De Datos | 37 |
| 2.4.2 | Bloque De Procesamiento De Datos | 37 |
| 2.4.3 | Tecnologías Aplicadas Al Subsistema De Toma Y Procesamiento De Datos | 38 |
| 2.4.3.1 | Raspberry PI..... | 38 |
| 2.4.3.2 | Módulos NFC PN532v3 | 38 |
| 2.4.3.3 | Etiquetas NFC | 39 |
| 2.4.3.4 | Base De Datos..... | 39 |
| 2.4.3.5 | Control RF Para Apertura De Brazos Mecánicos | 39 |
| 2.5 | Diseño De Subsistema De Visualización De Data | 39 |
| 2.5.1 | Tecnologías aplicadas al subsistema de visualización de datos..... | 40 |
| 2.5.2 | Bloque De Base De Datos | 40 |
| 2.5.3 | Bloque Servicio Web | 41 |
| 2.5.3.1 | Bloque Servidor Linux..... | 41 |
| 2.6 | Presentación De Tecnología Seleccionada | 42 |
| 3. | IMPLEMENTACIÓN Y PRUEBAS DEL PROTOTIPO | 42 |
| 3.1 | Implementación De Los Subsistemas | 43 |
| 3.1.1 | Implementación Del Subsistema De Toma Y Procesamiento De Datos | 43 |

| | |
|--|----|
| 3.1.1.1 Integración Y Configuración De Módulos Al Raspberry Pi | 43 |
| 3.1.1.2 Funcionamiento Del Sistema De Cobro Con Dinero Electrónico | 50 |
| 3.1.2 Implementación Del Subsistema De Cobro | 52 |
| 3.1.3 Implementación Del Subsistema De Visualización De Data | 58 |
| 3.1.3.1 Pago De Emergencia..... | 60 |
| 3.2 Pruebas Y Resultados | 61 |
| 3.2.1 Componentes De Hardware Y Software Para Pruebas De Funcionamiento | 62 |
| 3.2.2 Escenario De Pruebas..... | 64 |
| 3.2.3 Pruebas De Funcionalidad | 64 |
| 3.2.3.1 Fase De Ingreso Al Parqueadero | 65 |
| 3.2.3.2 Fase De Consulta Y Procesamiento De Datos | 66 |
| 3.2.3.3 Fase De Pago Emergente | 70 |
| 3.2.4 Resumen De Resultados Y Eventos..... | 73 |
| 3.2.5 Análisis De Resultados..... | 74 |
| 4. CONCLUSIONES Y RECOMENDACIONES..... | 77 |
| 4.1 Conclusiones..... | 77 |
| 4.2 Recomendaciones | 78 |
| Referencias..... | 80 |
| ANEXOS | 83 |

INDICE DE FIGURAS

| | |
|---|----|
| Figura 1. Evolución de formas de pago en Ecuador..... | 2 |
| Figura 2. Informática centralizada | 6 |
| Figura 3. Modelo cliente-servidor. | 6 |
| Figura 4. Modelo de aplicación capa 3..... | 7 |
| Figura 5. Componentes de red GSM..... | 8 |
| Figura 6. Auc Proceso de autenticación. | 9 |
| Figura 7. MS equipo móvil y tarjeta SIM..... | 12 |
| Figura 8. Diagrama módulo Sim900 para arduino..... | 13 |
| Figura 9. Diagrama módulo GSM USB. | 14 |
| Figura 10. Sesión USSD. | 17 |
| Figura 11. USSD llámame que no tengo saldo. | 18 |
| Figura 12. Raspberry PI | 20 |
| Figura 13. Módulo pn532 v3..... | 23 |
| Figura 14. Inducción de campo magnético..... | 27 |
| Figura 15. Comunicación inalámbrica por proximidad..... | 28 |
| Figura 16. Dispositivos NFC con alimentación propia. | 29 |
| Figura 17. Etiqueta NFC..... | 29 |
| Figura 18. Procesos a cumplir en el prototipo. | 31 |
| Figura 19. Flujo a cumplir arquitectura Del Prototipo | 32 |
| Figura 20. Esquema general de la estructura de cobro con dinero electrónico. | 33 |
| Figura 21. Diagrama de bloques del subsistema de cobro. | 35 |
| Figura 22. Diagrama de bloques del subsistema de toma y procesamiento de datos. | 37 |
| Figura 23. Subsistema de visualización de datos..... | 40 |
| Figura 24. Presentación de tecnología seleccionada..... | 42 |
| Figura 25. Conexión PN532v3 al Raspberry Pi..... | 44 |
| Figura 26. Flujo de entrada script in_parking.py..... | 45 |
| Figura 27. Flujo de salida script exit_parking.py..... | 46 |
| Figura 28. Acoplamiento módem GSM al Raspberry Pi. | 46 |
| Figura 29. Envío de cobro a la red GSM. | 47 |
| Figura 30. Esquema relacional de tablas. | 48 |
| Figura 31. Circuito acoplador con Raspberry PI..... | 50 |
| Figura 32. Flujo completo del sistema de cobro con dinero electrónico. | 51 |
| Figura 33. Flujo de envío de cobro por la red GSM..... | 53 |
| Figura 34. Envío mensaje USSD..... | 53 |

| | |
|---|----|
| Figura 35. Lectura de respuesta USSD en la red GSM..... | 54 |
| Figura 36. Diagrama de flujo escaneo USSD de red GSM. | 55 |
| Figura 37. Descripción grafica del punto 1 y 2. | 56 |
| Figura 38. Descripción grafica punto 3 y 4. | 56 |
| Figura 39. Descripción grafica punto 5..... | 56 |
| Figura 40. Graficas de preguntas punto 6. | 57 |
| Figura 41. Mensajes de texto de confirmación. | 57 |
| Figura 42. Grafica punto 8..... | 57 |
| Figura 43. Diferencia entre menú Persona Natural y Persona Natural con RUC. | 58 |
| Figura 44. Ingreso de matrícula en interface web de consulta. | 59 |
| Figura 45. Consulta de monto por pagar. | 59 |
| Figura 46. Acceso por usuario y contraseña. | 60 |
| Figura 47. Interface de pago manual..... | 61 |
| Figura 48. Pago manual exitoso..... | 61 |
| Figura 49. Terminal GSM para pruebas. | 63 |
| Figura 50. Lectura de etiqueta NFC y registro de ingreso en la BDD..... | 65 |
| Figura 51. Consulta de monto por pagar mediante la interface WEB..... | 66 |
| Figura 52. Registro de salida en la base de datos. | 66 |
| Figura 53. Proceso de envío de cobro por la red GSM. | 67 |
| Figura 54. Actualización de estado de pago enviado. | 67 |
| Figura 55. Mensaje USSD de solicitud de pago..... | 68 |
| Figura 56. Evidencia de cobro exitoso. | 68 |
| Figura 57. Notificaciones de pago exitoso al usuario por parte del BCE..... | 69 |
| Figura 58. Evidencia de cobro erróneo y reintento de cobro..... | 69 |
| Figura 59. Notificación de transacción fallida al usuario de parte del BCE. | 70 |
| Figura 60. Ingreso a la WEB administración. | 71 |
| Figura 61. Etiquetas NFC..... | 71 |
| Figura 62. Web de ingreso de identificación y pago emergente..... | 72 |
| Figura 63. Mensaje resultado del pago emergente. | 72 |
| Figura 64. Actualización de estado de pago. | 72 |
| Figura 65. Actualización de estado de pago. | 73 |
| Figura 66. Resultados. | 75 |
| Figura 67. Errores en cobro..... | 75 |
| Figura 68. Tiempo utilizado por intento. | 76 |

INDICE DE TABLAS

| | |
|--|----|
| Tabla 1. Interfaces tecnológicas que interactúan con el dinero electrónico. | 4 |
| Tabla 2. Códigos USSD más conocidos en el Ecuador. | 19 |
| Tabla 3. Puertos GPIO raspberry PI..... | 21 |
| Tabla 4. Hardware y software empleado en las pruebas del prototipo..... | 62 |
| Tabla 5. Características del terminal GSM usado en pruebas. | 63 |
| Tabla 6. Resumen de resultados..... | 73 |
| Tabla 7. Tiempos por prueba. | 74 |

1. FUNDAMENTO TEÓRICO.

El presente capítulo describe una reseña respecto al sistema de dinero electrónico en el Ecuador, regulaciones y funcionamiento. También se presenta una descripción de la tecnología GSM (Sistema global de comunicaciones móviles). Además describe características y funcionamiento de los módulos de Raspberry PI, los cuáles conformaran la plataforma para el prototipo propuesto. Posteriormente, se detallan las características y funcionamiento de la tecnología NFC (Comunicación de Campo Cercano).

1.1 Dinero Electrónico En El Ecuador

1.1.1 Antecedente

A lo largo del tiempo el Ecuador ha evolucionado las formas de pago, conforme a la época y a las necesidades de las personas. A continuación, se describe el proceso de su evolución cronológicamente:

- **Período Prehispánico:** Los pueblos ancestrales utilizaron medios de intercambio para el comercio (legal, 2015). Por de una moneda establecida se realizaba intercambio de productos como: collares de concha spondylus, hojas de coca, pepas de cacao y plumas de colores.
- **Período colonial:** Tras la llegada de los españoles, el Ecuador adoptó la onza o real como unidad monetaria.
- **Período Republicano:** En este periodo se da la aparición de los primeros bancos y junto a ellos el modo de pago con cheque el cual fue adoptado a finales del siglo XIX.
- **Período Republicano moderno:** El sucre es decretado como moneda oficial del Ecuador en 1884 pero no fue sino hasta 1927 tras la creación del Banco Central del Ecuador que se empiezan a emitir los primeros billetes de la moneda sucre. Diners club el primer medio electrónico de pago llega al ecuador en el año 1968. Un gran avance se da en 1972 con la

especialización de la banca Ecuatoriana en el desarrollo de tecnología informática bancaria con la implementación de cajeros automáticos.

De 1990 al año 2000, el sucre sufre una devaluación excesiva llegando a adoptarse el dólar como sistema monetario (legal, 2015). En 2002 se dan las primeras transferencias electrónicas bancarias, pero no es hasta el 2014 que se realiza el primer pago con dinero electrónico durante una demostración en el *Campus Party* en Quito.

Ecuador lleva tratando el tema de dinero electrónico desde el año 2011, donde la primera regulación acerca del dinero electrónico es emitida BCE (Banco Central del Ecuador), donde se encuentra descritos conceptos, obligaciones y responsabilidades para las personas naturales y jurídicas que hagan uso transaccional del Dinero Electrónico.

En la Figura 1 se muestra la evolución de las formas de pago en Ecuador desde el periodo prehispánico.

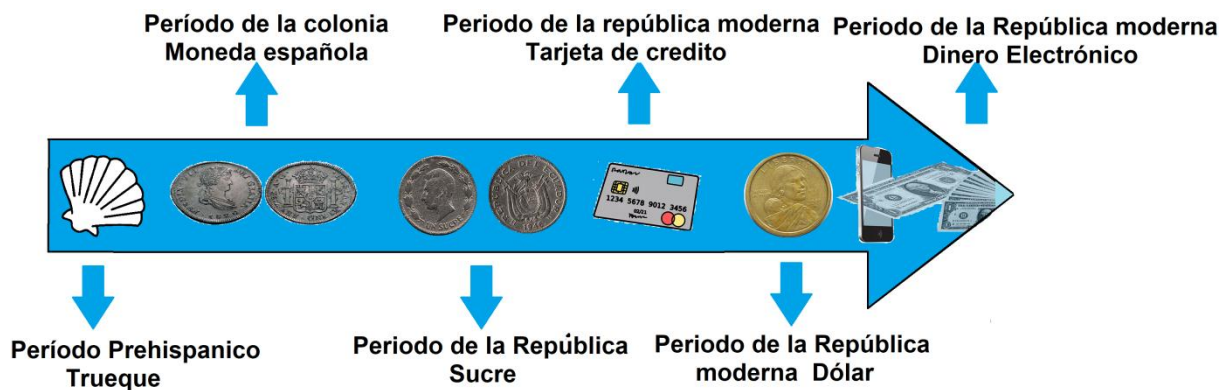


Figura 1. Evolución de formas de pago en Ecuador.

1.1.2 Definición De Dinero Electrónico

El dinero electrónico es un medio monetario que permite realizar cobros y pagos utilizando la moneda vigente en el Ecuador, mediante un sistema digital, sin la necesidad de contar con internet y una cuenta en una entidad financiera.

La regulación N°. 055 (Banco Central del Ecuador, 2014) emitida por el BCE el 28 de febrero de 2014 define al dinero electrónico:

“DINERO ELECTRONICO: es el valor monetario equivalente al valor expresado en la moneda de curso legal del país” (Banco Central del Ecuador, 2014), que cumple:

- a) “Se almacena e intercambia a través de dispositivos electrónicos, móviles, electromecánicos, fijos, tarjetas inteligentes, computadoras y otros, producto del avance tecnológico;” (Banco Central del Ecuador, 2014)
- b) “Es aceptado con poder liberatorio ilimitado y de libre circulación, reconocido como medio de pago por todos los Agentes Económicos en el Ecuador y para el pago de obligaciones públicas de conformidad con las normas que dicte el Organismo Regulatorio Competente;” (Banco Central del Ecuador, 2014)
- c) “Es convertible en efectivo a valor nominal; y,” (Banco Central del Ecuador, 2014)
- d) “Es emitido privativamente por el Banco Central del Ecuador sobre la base de las políticas y Regulaciones que expida el Organismo Regulatorio Competente y por ende se registra en el pasivo de la institución.” (Banco Central del Ecuador, 2014)

1.1.3 Principales Elementos En El Sistema De Dinero Electrónico

Los elementos esenciales que participan en el sistema de dinero electrónico son:

Emisor: BCE o la entidad bancaria de acuerdo a las leyes que lo faculten (Banco Central del Ecuador, 2014).

- Administrador: El BCE es el encargado de regir normas y estatutos sobre el funcionamiento del sistema de dinero electrónico (Banco Central del Ecuador, 2014).

- Entidades reguladoras y de control: Directorio del BCE, Arcotel, Superintendencia de Bancos (Banco Central del Ecuador, 2014).
- Operadores tecnológicos de telecomunicaciones: Las operadoras vigentes en el Ecuador en este caso (CNT, Claro, Movistar, Tuenti), las cuales interactúan tecnológicamente con la plataforma de dinero electrónico del BCE.
- Macro agentes: Entidades económicas públicas y privadas, que brinden atención al usuario que permita la recepción y transferencia con dinero electrónico; actualmente es BCE. Esto podría cambiar ya que se encuentra en negociación permitir a otras entidades bancarias el manejo de dinero electrónico. (Banco Central del Ecuador, 2014).
- Centros de transacción: Son los puntos de atención registrados para macro agentes, los cuales proveerán de productos y servicios del Sistema de Dinero Electrónico a los usuarios.
- Clientes: Son los usuarios (personas naturales o jurídicas, privadas o públicas) que tienen habilitada una cuenta de dinero electrónico en el Banco Central del Ecuador.

1.2 Tipos de interfaces tecnológicas que interactúan con dinero electrónico del BCE.

El Banco central del Ecuador realiza transacciones por diferentes tecnologías mencionadas en la Tabla 1.

Tabla 1.

Interfaces tecnológicas que interactúan con el dinero electrónico.

| Interfaces | Descripción |
|---|--|
| USSD (Servicio Suplementario de datos estructurados) | Es la interfaz principal que tiene el sistema de BCE para realizar todas las transacciones comerciales con Dinero Electrónico. USSD es un servicio de la telefonía móvil GSM (Sistema Global para Comunicaciones Móviles), |

| | |
|--|--|
| | actualmente la marcación *153# es utilizada para realizar creación de cuentas y movimientos en la misma. |
|--|--|

| | |
|--|--|
| SMS (Servicio de mensaje corto) | Es un servicio de la telefonía móvil GSM (Sistema Global para Comunicaciones Móviles), la plataforma de Dinero Electrónico del BCE utiliza SMS para notificar a los clientes las transacciones realizadas como carga, descarga, pago, cobro y las actividades realizadas por el cliente. |
|--|--|

| | |
|------------------------------------|--|
| APP (Aplicaciones móviles) | Software instalado en dispositivos móviles o Tablet creadas para una actividad específica, “Efectivo desde mi Celular” es la primera APP en el cual el cliente puede obtener información acerca del uso de Dinero Electrónico, lugares en donde cargar o descargar dinero, entre otras cosas. Esta APP está disponible en las tiendas <i>Google Play</i> y <i>App Store</i> . |
|------------------------------------|--|

| | |
|---|--|
| IVR (Respuesta de Voz Interactiva) | Es la gestión de acciones realizadas entre una central telefónica y el usuario. Las acciones son llevadas a cabo mediante pulsaciones DTMF (Sistema Multi frecuencial), se utiliza IVR en el caso de querer contactarse con un operador especializado en Dinero Electrónico del BCE, en Ecuador existe el contacto 1700-153-153 y la marcación directa 153 para contactarse con servicio al cliente del BCE. |
|---|--|

1.3 Nociones Fundamentales De Redes

1.3.1 La informática centralizada

A mediados del siglo pasado, los datos eran administrados en grandes sistemas u ordenadores centrales, se accedía a los mismos desde puestos externos (predecesores de los terminales). Los equipos de comunicación que permitían el intercambio de caracteres con el sistema central eran utilizados con pantallas y

teclados bastante simples. Compartir la información y servicios de la forma antes descrita fue el origen de las redes que conocemos hoy en día (Dordoigne, 2015).

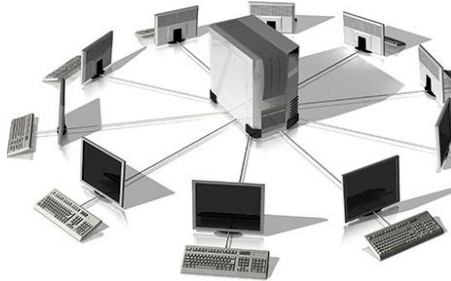


Figura 2. Informática centralizada.
Tomado de (Dakel, 2016)

1.3.2 La informática descentralizada

La creación de un nuevo enfoque en la gestión de la información: la información descentralizada fue posible con la llegada de los PC (Computador personal).

El primer PC ordenador personal aparece en 1981, el PC XT de IBM.

En la Figura 3 se muestra un modelo de informática descentralizada.

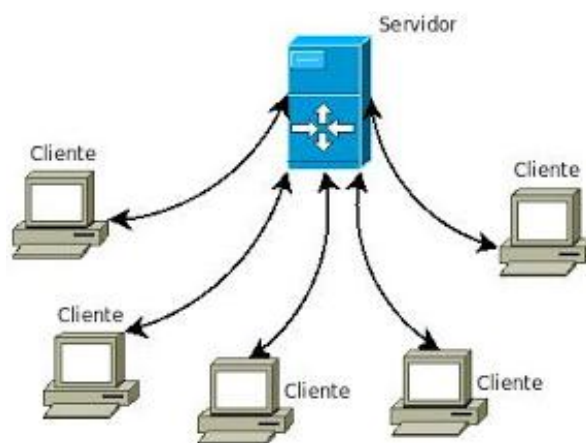


Figura 3. Modelo cliente-servidor.
Tomado de (Villegas, 2014)

Los servicios de red garantizan la distribución de la información, los mismos que permiten disponer en todas las estaciones de las mismas funcionalidades.

Con lo antes indicado aparece el concepto de redes pequeñas, posteriormente aparece el concepto de redes locales LAN (Dordoigne, 2015).

1.3.3 Redes heterogéneas

Red que conecta dos segmentos que tienen protocolos de enlace de datos diferentes mediante el uso puentes.

Es así que una red que consta de una Red en anillo y un segmento Ethernet se pueden conectar a fin de formar una sola red lógica. Organismos como el *Institute of Electrical and Electronics Engineers* (IEEE) proponen normas para los protocolos de capas físicas (IBM, 2017).

1.3.4 Redes informáticas actuales

En la actualidad, las redes se constituyen por ordenadores y sistemas operativos heterogéneos, que se interconectan a través de Internet.

A través de arquitecturas que incluyen diferentes capas (*tiers*, que significa capa), la distribución de recursos se multiplica. Los recursos de que dispone el usuario se emplean para organizar la información recibida cuando una capa intermedia administra las aplicaciones. Estas se han vuelto independientes de los datos y se distribuyen en distintos niveles (Dordoigne, 2015).

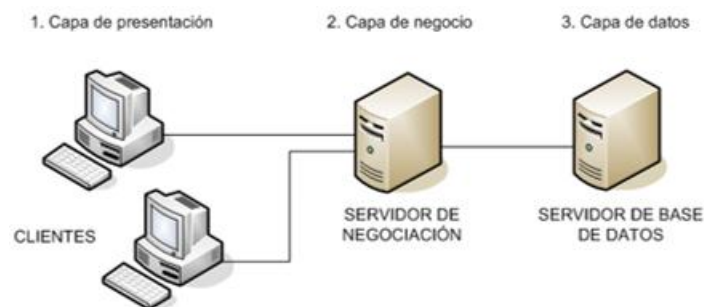


Figura 4. Modelo de aplicación capa 3.

1.4 Red GSM (*Sistema Global Para Comunicaciones Móviles*)

En este punto se describe las generalidades y arquitectura de la red GSM sobre la cual se ejecuta el servicio USSD (Servicio de Datos Suplementarios no Estructurados).

1.4.1 Generalidades De GSM

GSM es un sistema basado en celdas. Una celda se define como el área geográfica donde el operador mantiene cobertura, el área de cobertura de una celda es directamente proporcional a la potencia del trasmisor. Una cierta cantidad de frecuencias son empleadas en cada celda para el uso de un determinado número de usuarios. Además, con el fin de optimizar el espectro se utiliza reuso de frecuencias, para evitar solapamiento e interferencia debe existir una distancia considerable entre cada celda que haga reuso de frecuencia (Universidad Politecnica de Madrid, 2007).

1.4.2 Arquitectura De Una Red GSM

En la Figura 5, se resumen los elementos de una red GSM:

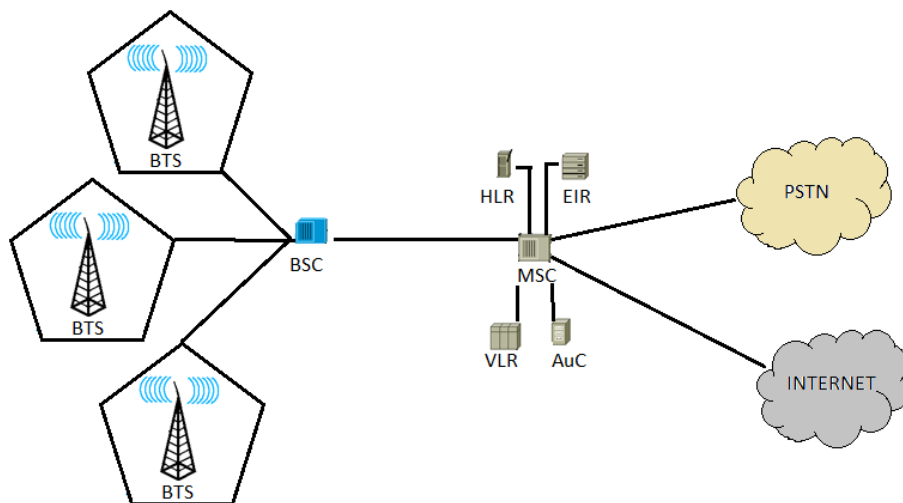


Figura 5. Componentes de red GSM.
Adaptado de (mrdesc, 2013)

1.4.3 Centro De Autenticación (Auc)

El centro de autenticación de usuario es encargado de cifrar la comunicación, identificación e integridad de los abonados dentro de la red GSM.

Con la utilización de tres elementos que constan de:

- *RAND*: Número aleatorio
- *SRES*: Respuesta
- *K_c*: Clave de cifrado

Estos tres elementos se utilizan para autenticar una llamada, y para obtener las claves de cifrado (Rey, 1998).

En la Figura 6 se especifica el proceso de autenticación.

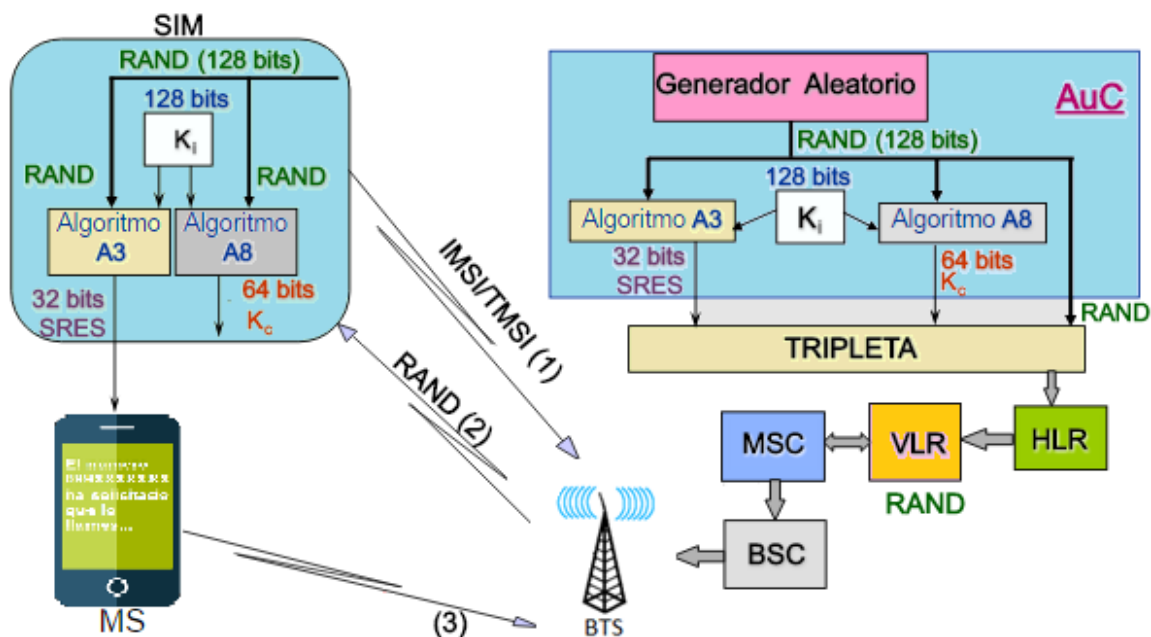


Figura 6. Auc Proceso de autenticación.
Adaptado de (Hernandez, 2016)

1.4.4 Registro De Identificación De Equipos (EIR)

El EIR hace el rol de almacenar listas de permiso de acceso a la red GSM, mediante la identificación de la identidad internacional de un equipo móvil IMEI.

Terminales móviles reportados como robados no podrán hacer uso de la red GSM gracias a esta lista de identificación de acceso.

1.4.5 Registro De Localización De Visitante (VLR)

Es una base de datos que almacena de manera temporal el registro de abonados a la red GSM, permitiéndoles conectarse con su HLR y hacer uso del servicio de telefonía móvil.

1.4.6 Registro De Ubicación Base (HLR)

Es una base de datos que almacena de manera permanente los datos de identidad y derechos de acceso a la red GSM. Entre los datos más importantes almacenados se encuentran (Rey, 1998):

- IMSI.
- Información de teleservicios y servicios portadores.
- Restricciones.
- Servicios suplementarios.
- Elementos Auc.

1.4.7 Centro De Comunicación Móvil (MSC)

El MSC, permite la localización y comunicación de los abonados en toda la red GSM con la ayuda de la información almacenada en las bases de datos HLR y VLR. Entre sus funciones más importantes se encuentran (Rey, 1998):

- Establecimiento, enrutamiento, control y terminación de las llamadas.
- Gestión de *handover* (traspaso de llamadas entre) entre centrales.
- Gestión de servicios suplementarios.
- Recolección de datos para tarificación.

1.4.8 Controlador De La Estación Base (BSC)

La BSC, administra los recursos de las radio bases y permiten la conmutación de los canales de radio por comandos remotos en las radio bases y estaciones móviles. Las principales funciones que realiza son (Rey, 1998)

- Gestión de canales de radio.
- Supervisión de las estaciones base.
- Traspaso de canales de la BSC.
- Trasmisión hacia las estaciones base.
- Localización de las estaciones móviles.

1.4.9 Estación Móvil (MS)

La Estación móvil del usuario consta de dos principales elementos para su funcionamiento los cuales son: el SIM (Módulo de la identificación del abonado) y el equipo móvil.

La tarjeta SIM es directamente relacionada al abonado, esta con el terminal móvil pertenecen al abonado. Un punto importante es la información que almacena la tarjeta SIM descrita a continuación:

- Número de serie.
- Clave de algoritmo de cifrado (Kc).
- Identificación Internacional del abonado móvil (IMSI).
- Algoritmo de cifrado (A5).
- Identificación temporal del abonado móvil (TMSI).
- Clave de algoritmo de autenticación (Ki).
- PIN (Clave corta de desbloqueo).
- PUK (Clave larga de desbloqueo).

Además, brinda memoria EEPROM (Memoria de solo lectura programable y borrrable electrónicamente) la cual permite el uso de los siguientes servicios:

- Agenda con últimos números marcados
- Agenda personal de teléfonos
- Lista de redes preferidas por el cliente en caso de *roaming*
- Almacenamiento de mensajes cortos (SMS)

En la Figura 7 se aprecia un equipo móvil junto a su tarjeta SIM.



Figura 7. MS equipo móvil y tarjeta SIM.
Adaptado de (Selena, 2017)

1.4.10 Módulos GSM

Un módulo GSM es un dispositivo compacto cuyo objetivo es intercomunicarse con la radio base, permitiendo el acceso a la red de servicio de telecomunicación. Estos pueden trabajar en frecuencias de 850/900/180/1900 MHz, permitiendo llamadas de voz, envío de mensajes, acceso al servicio USSD y todas las funcionalidades de la tecnología GSM.

Este dispositivo será utilizado para enviar mensajes USSD en el sistema de pago.

Existen diferentes tipos de módulos GSM. A continuación se mencionan dos, cuyas funciones son muy similares.

- Módulo GSM por conexión de pines SIM900: Permite acoplarse a plataformas como Arduino, Raspberry PI o cualquier placa que tenga la habilidad de utilizar puertos GPIO. En la Figura 8 se puede observar un diagrama de la estructura de este módulo.

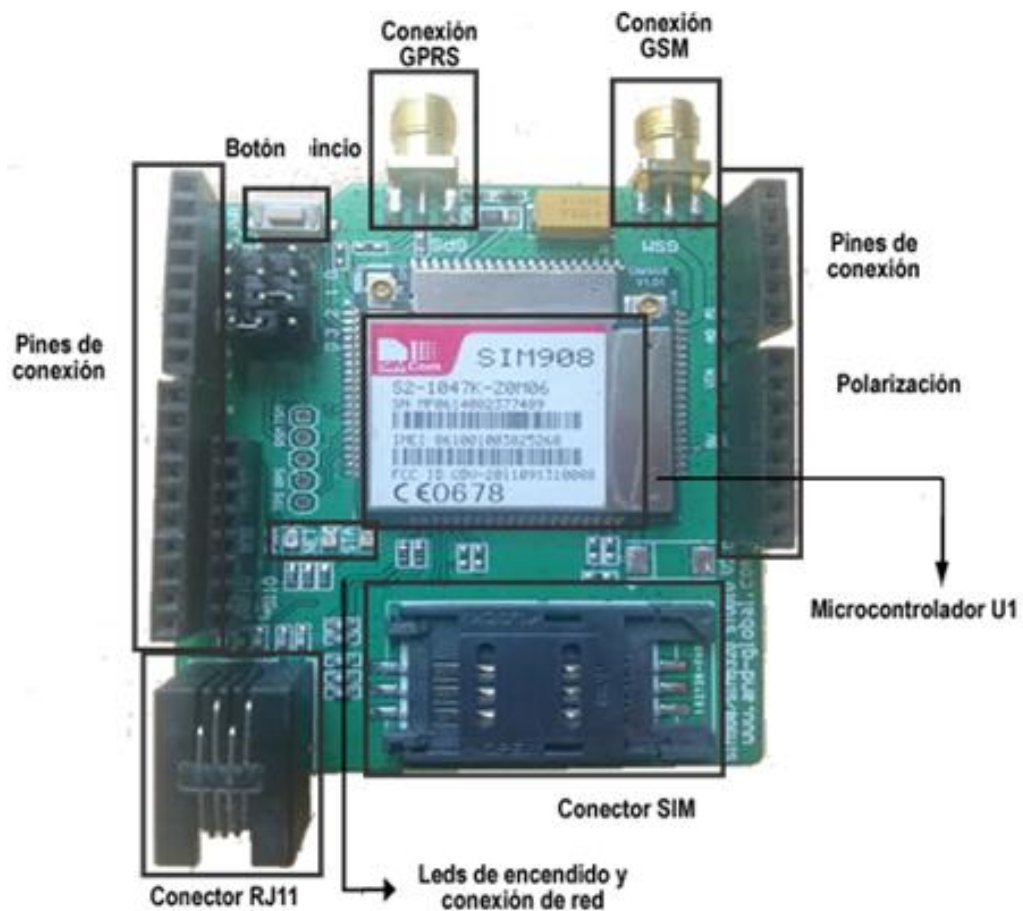


Figura 8. Diagrama módulo Sim900 para arduino.

- Módulo GSM USB: Permite la conectividad a la red celular mediante la conexión de puertos USB. En la Figura 9 se puede observar un diagrama de la estructura del módulo.

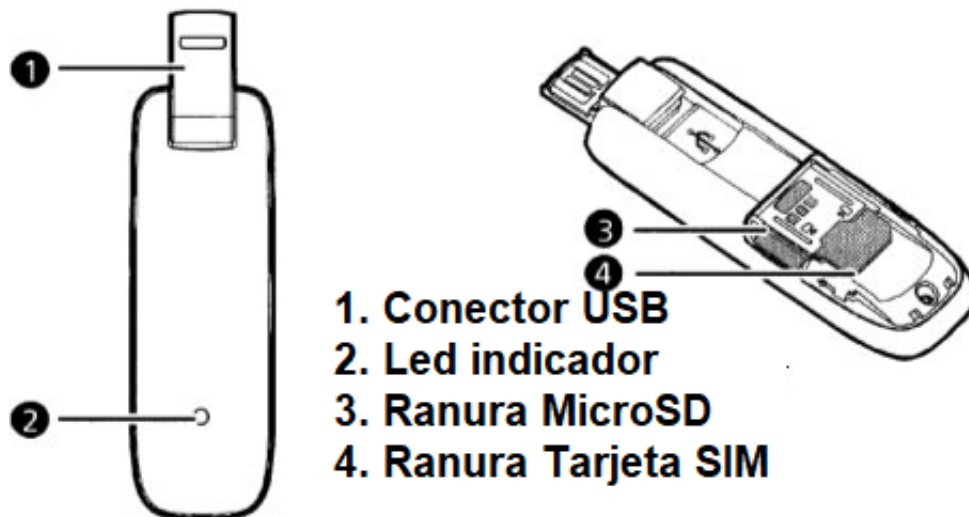


Figura 9. Diagrama módulo GSM USB.
Adaptado de (Huawei, 2016)

1.5 Tecnología USSD (Servicio De Datos Suplementarios No Estructurados)

USSD se encuentra dentro del estándar GSM02.90 USSD fase 1 (ETSI, 1997) y GSM 03.90 (ETSI, 1997). Teniendo un comportamiento diferenciado al servicio de mensaje corto SMS, el cual funciona almacenando mensajes y enviándolos. USSD funciona de un modo conversacional, es decir, el cliente inicia una sesión, recibe una respuesta inmediata y puede responder nuevamente una y otra vez hasta que el abonado, aplicación o por tiempo de espera agotado la conexión es liberada.

USSD es un servicio cuya información es transmitida a través de los canales de señalización de la red GSM. Gracias a USSD las operadoras pueden brindar mejores prestaciones de Teleservicios básicos.

1.5.1 Servicios Prestados De Red

Estos servicios se encuentran clasificados en:

- Servicios básicos
 - Servicios portadores
 - Teleservicios
- Servicios suplementarios

1.5.2 Servicios Portadores

Estos servicios permiten al usuario la transferencia de datos, estos son enviados por conmutación de circuitos, además son caracterizados por:

- Modo de transferencia de información: Conmutación de circuitos
- Establecimiento de comunicación: Por demanda
- Simetría de la comunicación: Bidireccional simétrica
- Configuración de la comunicación: Punto a punto. (Balboa, 2002)

1.5.3 Teleservicios

Teleservicios se llama aquellos servicios que brindan la capacidad de comunicación entre abonados, mediante protocolos preestablecidos.

Una red GSM soporte los siguientes teleservicios:

- Telefonía
- Llamadas de emergencia.
- Servicio de mensajes cortos.
- Servicio de voz para grupo.
- Servicio de tarifación.
- Servicio de plan privado de numeración. (Balboa, 2002)

1.5.4 Servicios Suplementarios

Se los denomina servicios suplementarios debido a que complementan el servicio básico de telecomunicaciones, utilizado por los MS, estos se implementan sobre los teleservicios, son los siguientes:

- Reenvío de llamadas.
- Bloqueo de llamadas.
- Identificación de la línea conectada.
- Llamada en espera.
- Servicios de datos suplementarios no estructurados.
- Operador de restricción determinado. (Balboa, 2002)

1.5.5 Funcionamiento Y Características

El servicio USSD permite un acceso sencillo a servicios suplementarios de telecomunicaciones. Una transacción USSD empieza cuando el abonado envía un mensaje USSD, mediante la red GSM hacia el *Gateway* USSD, el cual luego de interpretar el código permite la interacción con la aplicación deseada.

Características importantes del servicio USSD:

- Cualquier dispositivo GSM puede acceder al servicio de USSD, no es necesario un teléfono inteligente.
- Puesto que este servicio no necesariamente requiere almacenar información para enviarla los tiempos de respuesta son menores comparados al servicio SMS.
- Los servicios USSD locales pueden ser accedidos también desde el uso de *roaming*.
- USSD tiene la capacidad de manejar mensajes de texto hasta 182 caracteres o 160 *bytes*. (Balboa, 2002).

Los Datos suplementarios no estructurados (USSD) es una forma de transmitir instrucciones o información a través de los canales de señalización de la red GSM. Si un cliente GSM requiere acceder a un servicio debe realizar el proceso de inicio de sesión el cual se puede apreciar en la Figura 10 que permanece abierta durante la utilización de la aplicación, esta conexión puede ser liberada por el usuario o cuando el tiempo de espera máximo expira.

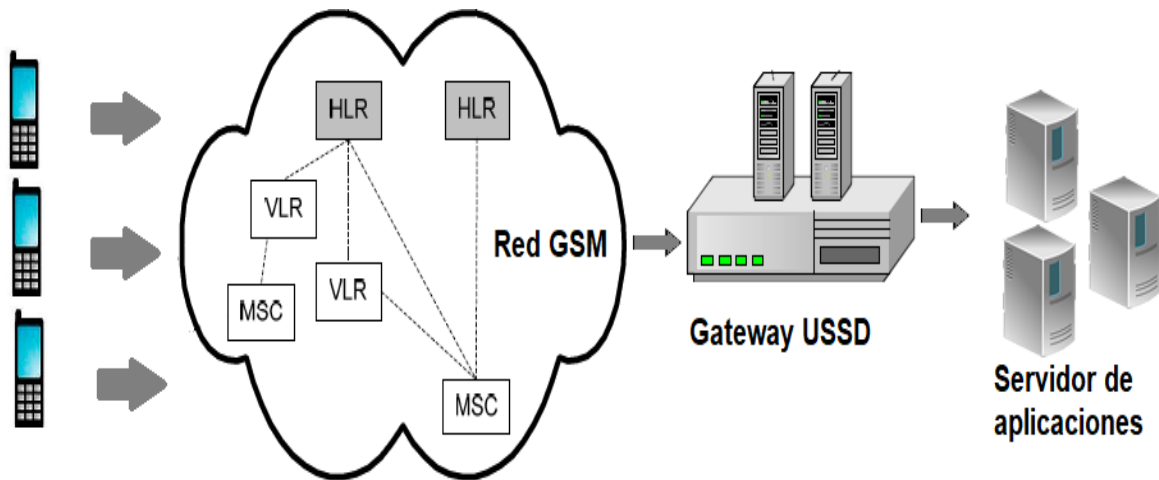


Figura 10. Sesión USSD.

Existen dos modos de operación USSD:

- Modo hombre maquina (MMI).
- Modo aplicación.

1.5.6 MMI (Modo Hombre Maquina)

Es utilizado para transportar cadenas de caracteres desde el abonado hacia la red GSM, y respuestas desde la red GSM hacia el abonado. En la cadena de caracteres enviada es permitido incluir números anteponiendo los signos “*” y/o “#”, las transacciones USSD pueden ser:

Iniciadas por el móvil:

- Operación inicial en la estación móvil: Cuando el usuario ingresa una cadena MMI (*153#).
- Operación en la Red: La red GSM analiza el contenido de la cadena, si esta cadena es válida realiza la acción correspondiente, como entregar información del terminal móvil o realizar una acción, como pago de

servicios. Si la cadena no es válida, se notifica a la estación móvil con un mensaje de error.

Iniciadas por la red:

- Operación inicial en la red: La red envía al terminal móvil cadenas no estructuradas. Las cuales contienen información determinada por el operador de telefonía móvil.
- Operación en el terminal móvil: Si la cadena no es recibida por el terminal se envía una señal de error al nodo que inició la operación. La red puede solicitar una respuesta o simplemente desplegar información para el usuario. Uno de los mensajes no estructurados más conocidos es el “llámame que no tengo saldo” véase en la Figura 11.

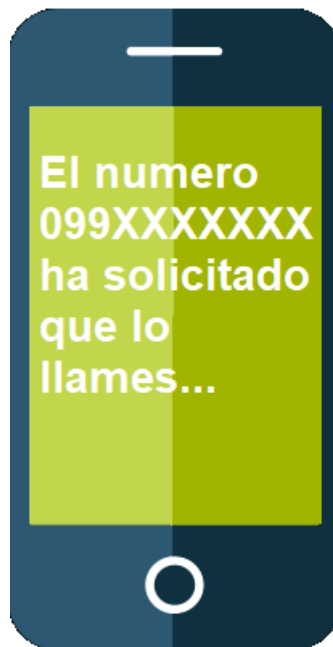


Figura 11. USSD llámame que no tengo saldo.

1.5.7 Modo Aplicación

Los mensajes no estructurados USSD soportan la comunicación entre la aplicación de la MS y la aplicación correspondiente en la red. Permite la

transferencia transparente de datos binarios entre la red y el terminal móvil. Las aplicaciones pueden utilizar mensajes no estructurados durante una llamada o fuera de ella.

1.5.8 Códigos USSD Más Conocidos En El Ecuador

Las operadoras de telecomunicaciones utilizan el servicio USSD para brindar cierto tipo de acciones, los cuales se mencionan en la .

Tabla 2.

Tabla 2.

Códigos USSD más conocidos en el Ecuador.

| Operadora | Código USSD | Descripción |
|------------------|--------------------|---|
| Conecel S.A. | *123# | Consulta de saldos, traspaso de saldo, servicios suplementarios. |
| Conecel S.A. | *102# | Consulta de saldo, megas o saldo promocional |
| Otecel S.A. | *100# | Consulta de saldos, traspaso de saldo, servicios suplementarios. |
| Otecel S.A. | *120# | Consulta de saldos, traspaso de saldo, servicios suplementarios. |
| Otecel S.A. | *104*099*****# | Se envía mensaje USSD solicitando que lo llame "llámame que no tengo saldo" |

| | | |
|--------|-------|--|
| CNT EP | *611# | Consulta de saldos, traspaso de saldo, servicios suplementarios. |
|--------|-------|--|

1.6 Microcomputadores

Los microcomputadores que cumplen con la función de un ordenador, perfectamente funcional sin todos sus periféricos extras. Entre este tipo de placas podemos encontrar al Raspberry PI el cual es el utilizado en este proyecto, Orange PI, Cubie Boards entre otros.

1.6.1 Raspberry PI

Raspberry PI es una placa computadora de bajo coste (RASPBerry PI FOUNDATION, 2017). Se puede decir que es un computador del tamaño de una tarjeta de crédito, tal como se muestra en la Figura 12. Fue creado en el Reino Unido por la Fundación Raspberry PI en la Universidad de Cambridge.

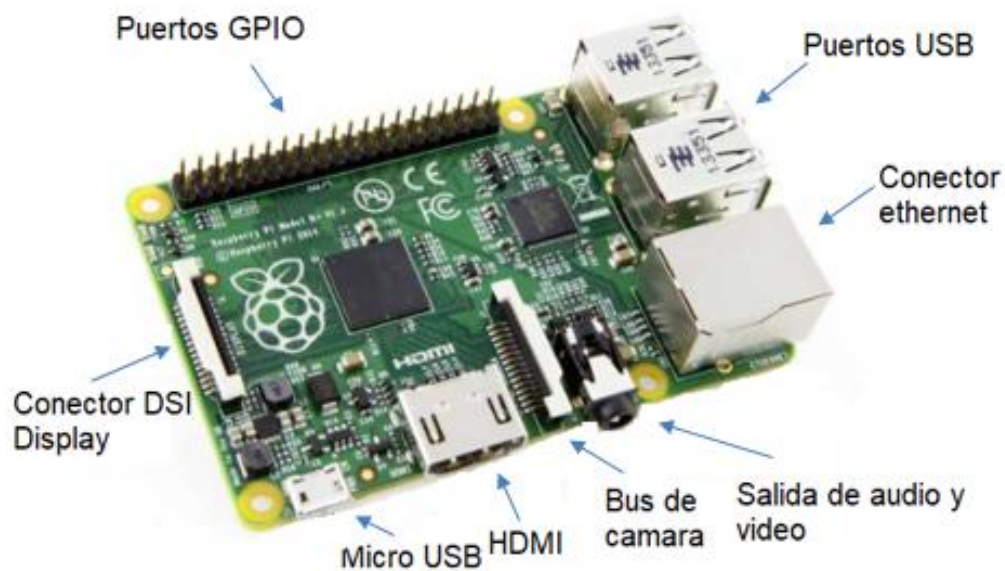


Figura 12. Raspberry PI

1.6.2 Principales Elementos Del Raspberry PI

Los principales elementos que conforman y permiten el funcionamiento del Raspberry *PI* son los siguientes:

- **CPU:** Microprocesador basado en ARM.
- **GPU:** Utiliza una tarjeta *dual core* video multimedia, la cual es capaz de reproducir contenidos multimedia de alta calidad.
- **RAM:** La memoria RAM se encuentra embebida en la placa y va desde 252 MB hasta 1024 en su última versión.
- **Almacenamiento:** El raspberry *PI* no cuenta con un disco duro tradicional, por eso dispone de una ranura para memorias SD, en la cual será almacenado todo el sistema operativo.
- **Salidas de video:** Dependiendo de la versión puede contar con un conector RCA, VGA o HDMI.
- **Bus USB:** Dependiendo de la versión puede contar desde 1 puerto hasta 4 puertos USB 2.0.
- **Tarjeta de RED:** Cuenta con un conector RJ-45 que nos proporciona una conectividad a 10/100 Mbps.
- **Energía y Alimentación:** Cuenta con un conector *microUSB* estándar de 5v. El consumo mínimo requerido es de 700mA.
- **Pines de entrada y salida “GPIO”:** Cuenta con diferentes pines de entrada y salida los cuales se encuentran descritos en la Tabla 3.

Tabla 3.
Puertos GPIO raspberry PI.

| Pin# | Nombre | Nombre | Pin# |
|------|--------------------|--------|------|
| 1 | 3.3v DC | 5v DC | 2 |
| 3 | GPIO02 (sda1, I2C) | 5v DC | 4 |

| | | | |
|-----------|-------------------|--------------|-----------|
| 5 | GPIO03(scl1, I2C) | GROUND | 6 |
| 7 | GPIO4(GPIO_GCLK) | GPIO14 | 8 |
| 9 | TIERRA | GPIO15 | 10 |
| 11 | GPIO017 | GPIO18 | 12 |
| 13 | GPIO027 | TIERRA | 14 |
| 15 | GPIO22 | GPIO23 | 16 |
| 17 | 3.3 v DC | GPIO24 | 18 |
| 19 | GPIO10 | TIERRA | 20 |
| 21 | GPIO09 | GPIO25 | 22 |
| 23 | GPIO11 | GPIO08 | 24 |
| 25 | TIERRA | GPIO07 | 26 |
| 27 | ID_SD(I2C) | ID_SC (I2C) | 28 |
| 29 | IGPIO05 | GROUND | 30 |
| 31 | GPIO06 | GPIO12 | 32 |
| 33 | GPIO13 | TIERRA | 34 |
| 35 | GPIO19 | GPIO16 | 36 |
| 37 | GPIO26 | GPIO20 | 38 |
| 39 | TIERRA | GPIO21 | 40 |

Nota. DC = corriente continua; GPIO = propósito general entrada salida; I2C = circuito inter integrado.

1.6.3 Módulo NFC/RFID

NFC/RFID PN532 a diferencia de módulos anteriores es muy pequeño y práctico como se muestra en la Figura 13. Este módulo es utilizado para aplicaciones que trabajan en la banda de frecuencia 13.56 MHz. Además soporta I2C, SPI y HSU, permitiendo la lectura y escritura de las etiquetas NFC (Wilson, 2013).

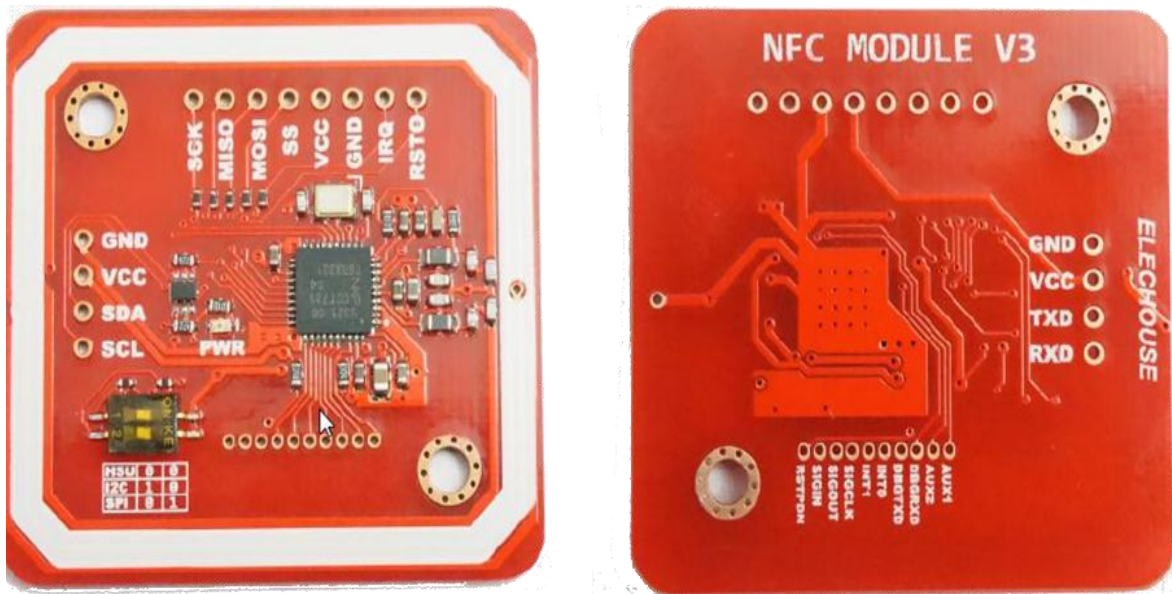


Figura 13. Módulo pn532 v3

1.7 Tecnologías Y Herramientas

Este subcapítulo describe herramientas y tecnologías que generalmente son utilizadas para solventar algún tipo de necesidad, como por ejemplo una base de datos, que nos sirve para almacenar información de manera ordenada.

1.7.1 Bases De Datos Sql (Lenguaje De Consultas Estructuradas)

Es un repositorio de información almacenada sistemáticamente que pertenece a un mismo contexto, la cual puede estar relacionada entre sí, además puede ser consultada simultáneamente por varios usuarios, utiliza lenguaje de consultas estructuradas SQL que permite la explotar la flexibilidad y potencia de los sistemas

de datos relacionales. Existen diferentes proyectos de base de datos SQL entre los más conocidos se encuentran ORACLE data base, Microsoft sql server, Mysql, (IBM, 2017)

1.7.1.1 MaríaDB

MaríaDB es un sistema de gestión de base de datos SQL derivado de MySQL desarrollado por la comunidad de software libre que se encuentra disponible bajo los términos de la licencia GPL v2. (mariadb, 2017). Comúnmente MariaDB es encontrado trabajando en conjunto con aplicaciones como servidores web con consultas dinámicas, almacenamiento de información dinámica que es el caso que utilizaremos en este proyecto.

1.7.2 Html (Lenguaje De Marcas De Hipertexto)

HTML es utilizado comúnmente para presentación de páginas web. Es un protocolo perteneciente a la capa aplicación.

Se basa en *tags* o marcas que sirven para estructurar documentos, pero no describe o define su apariencia. La estructura básica de este lenguaje cuenta con dos secciones la cabecera, donde se encuentra definidas propiedades del documento y el cuerpo que es la sección que contiene los elementos que serán desplegados la página web. (Vertice, 2009)

1.7.3 Php (Procesador De Hipertexto)

PHP es un lenguaje de programación de código abierto orientado a objetos bastante popular cuando se trata de desarrollos WEB, ya que permite tener un mejor control de HTML y realizar varias funciones que HTML no permite como por ejemplo inicio de sesiones y consultas a bases de datos (PHP, 2017).

1.7.4 Python

Python es un lenguaje de programación de código abierto orientado a objetos, es un lenguaje poderoso y sencillo de utilizar utiliza una sintaxis elegante que facilita la lectura de los programas creados con él. Esto hace que Python sea ideal para el desarrollo de prototipos.

1.8 Sistemas Operativos

Un sistema operativo es un conjunto de programas que permite interacción con el hardware de la computadora (Wolf, 2015). Sus principales funciones son:

- **Abstracción:** Los programas se encuentran ejecutándose dentro del sistema operativo por lo que estos no deben preocuparse por tener acceso directo al hardware. El sistema operativo es el encargado de brindar una serie de abstracciones para que los desarrolladores puedan enfocarse a solventar las necesidades de los usuarios.
- **Administración de recursos:** Un ordenador o sistema computarizado tiene una cierta cantidad de recursos (almacenamiento, memoria, tiempo de procesamiento, procesamiento grafico etc.) los cuales deben ser asignados de una manera efectiva a cada uno de los procesos ejecutados.
- **Aislamiento:** Actualmente todos los sistemas operativos son diseñados con un fin multitarea y multiusuario, en el cual los usuarios no deben preocuparse por procesos ejecutados no necesariamente por él, brindando una experiencia dedicada a cada usuario.

Existen varios diferentes sistemas operativos entre los más populares sistemas operativos para computadores se encuentran:

1.8.1 WINDOWS

Windows pertenece a la gama de software comercial, es decir para hacer uso de él se requiere adquirir una licencia. Este sistema operativo es desarrollado y

comercializado por Microsoft. Es muy popular por su sencillez al momento de interactuar con el usuario. Existen varias versiones de Microsoft Windows que van desde versiones de hogar hasta la versión corporativa para servidores.

1.8.2 MacOS

MacOS es un sistema operativo basado en Unix, desarrollado y comercializado por Apple. Su mayor popularidad reside en esta desarrollado específicamente para el hardware que hace uso y viceversa, por lo que utiliza al máximo todos sus recursos, además de contar con una interfaz muy amigable y estética.

1.8.3 GNU/Linux

Es un sistema operativo de software libre basado en Unix, del cual derivan diferentes distribuciones enfocadas a diferentes tipos de uso, como ejemplo: Ubuntu, una distribución basada en debían mayormente conocida por su versión de escritorio.

1.8.4 Raspbian

Es un sistema operativo basado en debían, que permite utilizar al 100% los recursos de un Raspberry PI. Raspbian es desarrollado por la fundación Raspberry PI, es catalogado como sistema operativo nativo para Raspberry PI (RASPBERRY PI FOUNDATION, 2017).

1.9 Tecnología NFC (Comunicación De Campo Cercano)

1.9.1 Definición De La Tecnología NFC

NFC se define como una tecnología inalámbrica de corto alcance la cual permite el intercambio de datos de una manera bidireccional y con un alcance aproximado de 10 cm. Esta tecnología cuenta con el estándar ISO 18092 (ISO/IEC, 2013).

Los dispositivos que utilizan tecnología NFC pueden comunicarse con cualquier lector y tarjeta inteligente especificadas dentro del estándar ISO/IEC 14443 (ISO/IEC, 2002).

1.9.2 Especificaciones Técnicas

NFC se comunica vía inducción de campo magnético, donde dos lazos son localizados dentro de cada campo cercano como se muestra en Figura 14.



Figura 14. Inducción de campo magnético.

Los dispositivos pueden transmitir y recibir datos al mismo tiempo, por lo que su comunicación es bidireccional, a continuación se detalla las características técnicas de esta tecnología:

- **Comunicación inalámbrica por proximidad:** La inducción electromagnética es utilizada para este tipo de comunicación. Las antenas de los dispositivos compatibles con esta tecnología generan un campo electromagnético de radiofrecuencia como se observa en la Figura 15. Uno de los dispositivos inicia la comunicación cuando se encuentra dentro del alcance de este campo magnético. Además, el alcance de este campo es

reducido teóricamente entre los 10 y 20 cm, pero en la práctica alcanza una distancia de hasta 4 cm (NFC Forum, 2010).

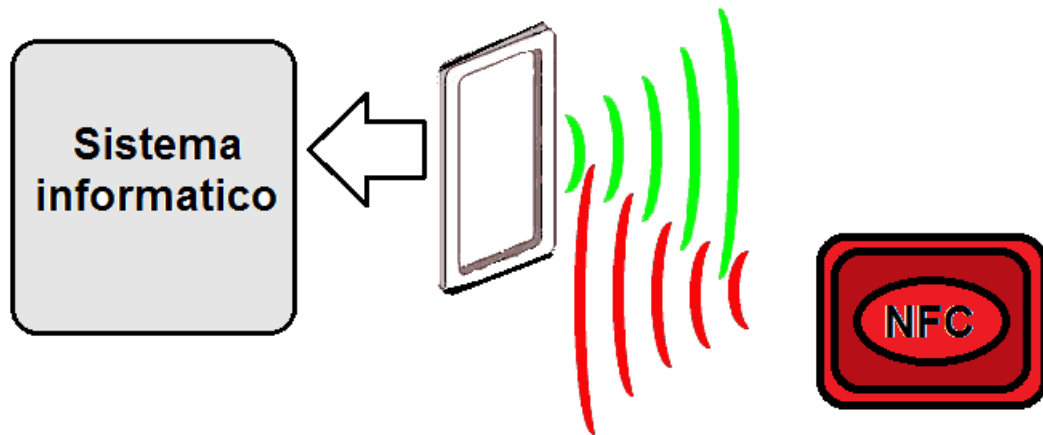


Figura 15. Comunicación inalámbrica por proximidad.

- **Transacciones de datos:** Las tasas de transmisión para comunicaciones NFC son pequeñas oscilan entre los 106 a 848 kbits/s. Debido a la baja tasa de transmisión de la tecnología NFC esta no es orientada a la transmisión masiva de datos, más bien a pequeñas transacciones entre dispositivos compatibles (NFC Forum, 2010).
- **Banda de radiofrecuencia ISM (*Industrial, Scientific and Medical*):** La tecnología NFC utiliza una frecuencia de 13.56 MHz. Esta banda pertenece a un conjunto de radiofrecuencia ISM, utilizada con fines industriales, científicos y médicos. Esta banda de frecuencia no requiere licencia para su uso, por lo que la transmisión de datos mediante NFC no tiene costo de uso (NFC Forum, 2010).

1.9.3 Tipos Y Modos De Comunicación

Existen dos tipos básicos de comunicación detallados a continuación:

- **Activo:** Se denomina tipo de comunicación activa, cuando el dispositivo tiene la capacidad de crear su propio campo de radiofrecuencia, para esto es necesario alimentación eléctrica, generalmente baterías o pilas como

celulares o dispositivos específicos como se muestra en la Figura 16. (NFC, 2012).



Figura 16. Dispositivos NFC con alimentación propia.

- **Pasivo:** Se denomina comunicación pasiva, cuando el dispositivo recibe alimentación del campo de radiofrecuencia de otro dispositivo como son las etiquetas NFC como se observa en la Figura 17.



Figura 17. Etiqueta NFC

Dependiendo del tipo de comunicación se definen los modos o roles que puede tomar el dispositivo NFC. La comunicación es basada en el principio de

pregunta responde, pero un dispositivo responde a otro solamente si anteriormente se ha iniciado una comunicación previa. Los roles requeridos para la comunicación NFC son:

- **Iniciador:** Como su nombre lo indica, es el dispositivo que inicio la comunicación, en el caso de ser una comunicación bidireccional, el primer dispositivo que inicie la comunicación es considerado como iniciador hasta que la comunicación termine.
- **Objetivo:** Es el que responde la comunicación al iniciador.

De acuerdo a las definiciones descritas, es posible deducir que existen combinaciones de tipos y roles que no son compatibles. Como ejemplo, un dispositivo activo puede tomar el rol de iniciador u objetivo, pero, un dispositivo pasivo no es capaz de generar su propio campo de radiofrecuencia, por lo que no puede ser iniciador (NFC, 2012).

1.9.4 Limitaciones NFC

Como toda tecnología inalámbrica NFC tiene limitaciones en su comunicación. La limitación más notable radica en que su comunicación es *unicast* (único receptor, único transmisor), es decir, un dispositivo activo no es capaz de comunicarse con varios dispositivos pasivos al mismo tiempo.

2. DISEÑO DEL PROTOTIPO

El presente capítulo describirá el diseño del prototipo propuesto, sus principios de funcionamiento, dispositivos utilizados y software empleado.

2.1 Requerimientos Del Sistema

El flujo general del prototipo se presenta en la Figura 18.

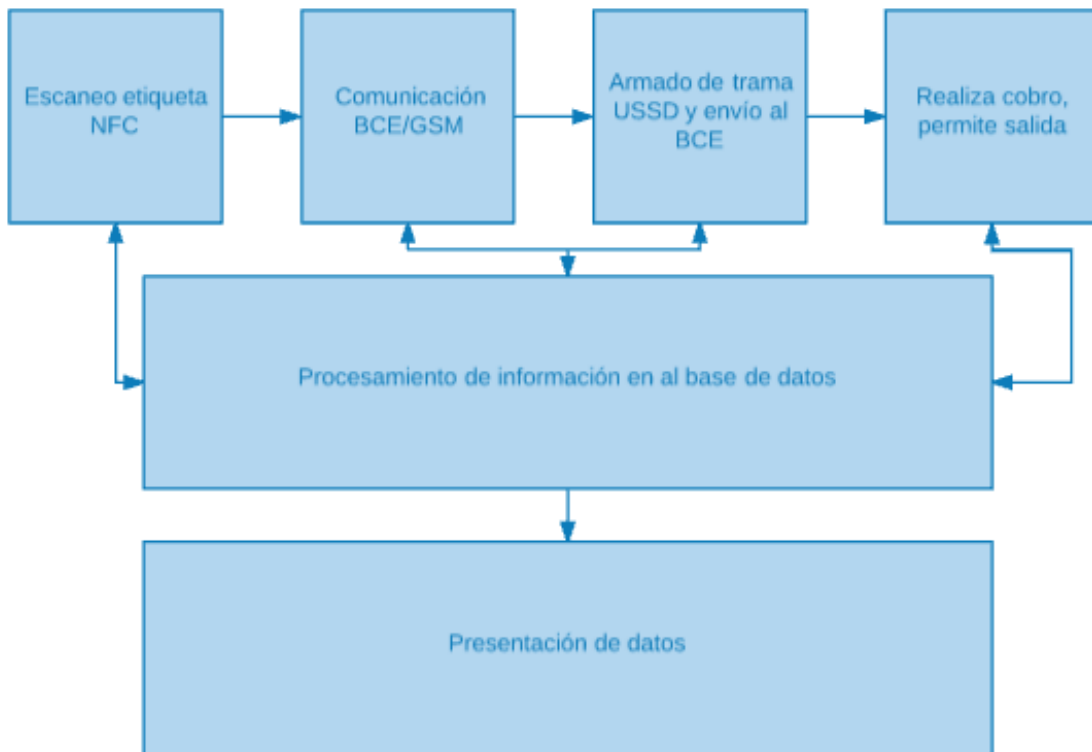


Figura 18. Procesos a cumplir en el prototipo.

Los requerimientos que deben ser cumplidos en el prototipo son:

- a) Comunicación con el BCE mediante GSM.
- b) Toma de etiqueta mediante escaneo NFC.
- c) Tabulación de tiempo de estancia.
- d) Envío de cobro a la plataforma de BCE para debito del costo por tiempo tabulado.
- e) Consulta del costo por el tiempo transcurrido.
- f) Salida del establecimiento.

El flujo que permite cumplir los requerimientos planteados se encuentra descrito en la Figura 19.

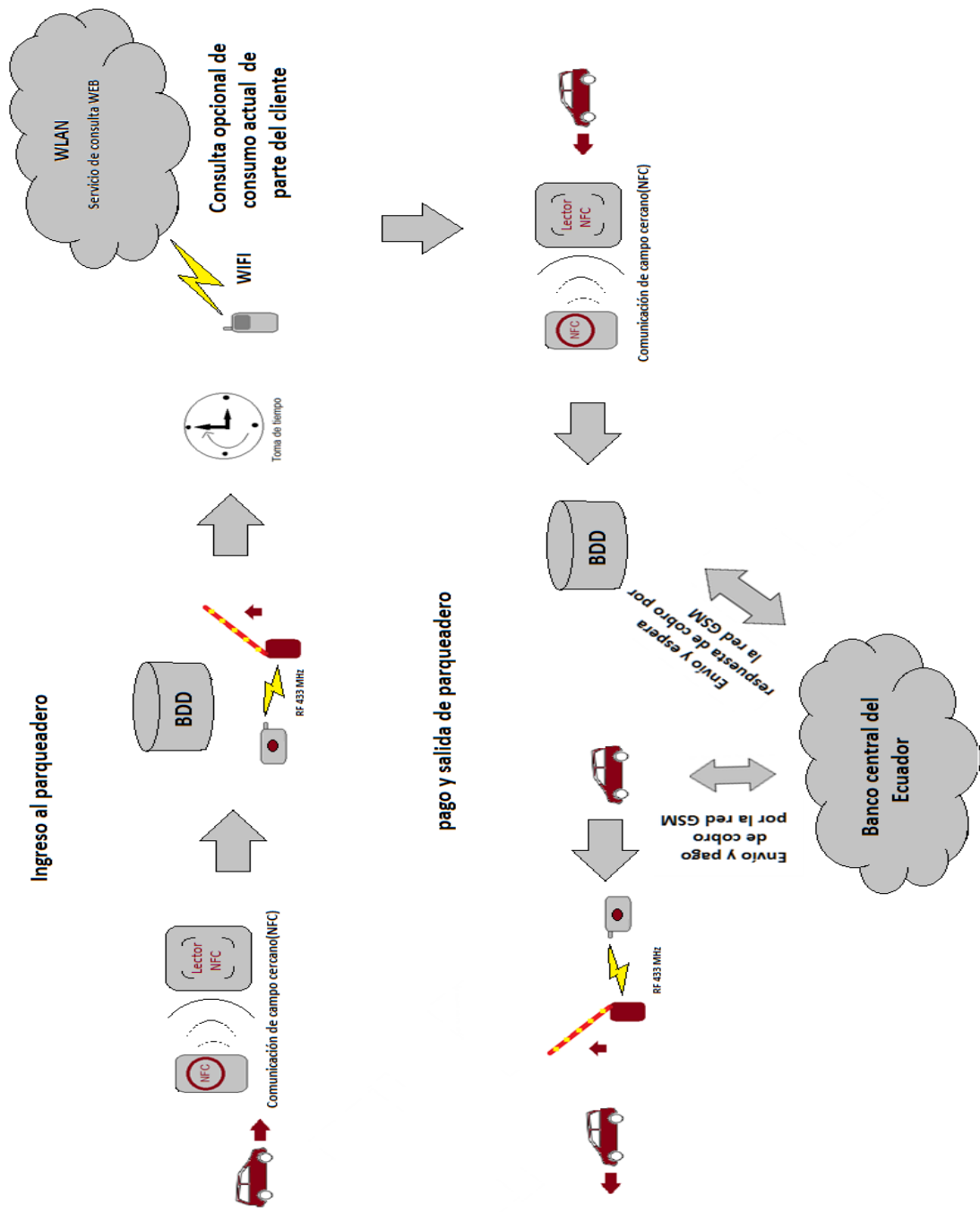


Figura 19. Flujo a cumplir arquitectura Del Prototipo

El presente prototipo propuesto está compuesto por varios subsistemas, cada uno cumple con funciones específicas con el fin de permitir la entrada, salida al parqueadero, visualización de datos y cobro del parqueadero, las cuales interactúan entre ellas para cumplir el objetivo de registrar y permitir el ingreso al parqueadero con la ayuda de la tecnología NFC, la tabulación del costo por la estancia, en el parqueadero, posteriormente permitir al cliente la consulta del valor a pagar y finalmente realizar el cobro mediante dinero electrónico permitiendo la salida del establecimiento. Como alternativa al cobro con dinero electrónico se presenta un sistema de cobro emergente que consiste en permitir el pago de manera tradicional con dinero físico y mediante una página WEB registrar el pago en el sistema completando el flujo de cobro.

A continuación en Figura 20, se presenta un esquema general de la estructura del cobro.

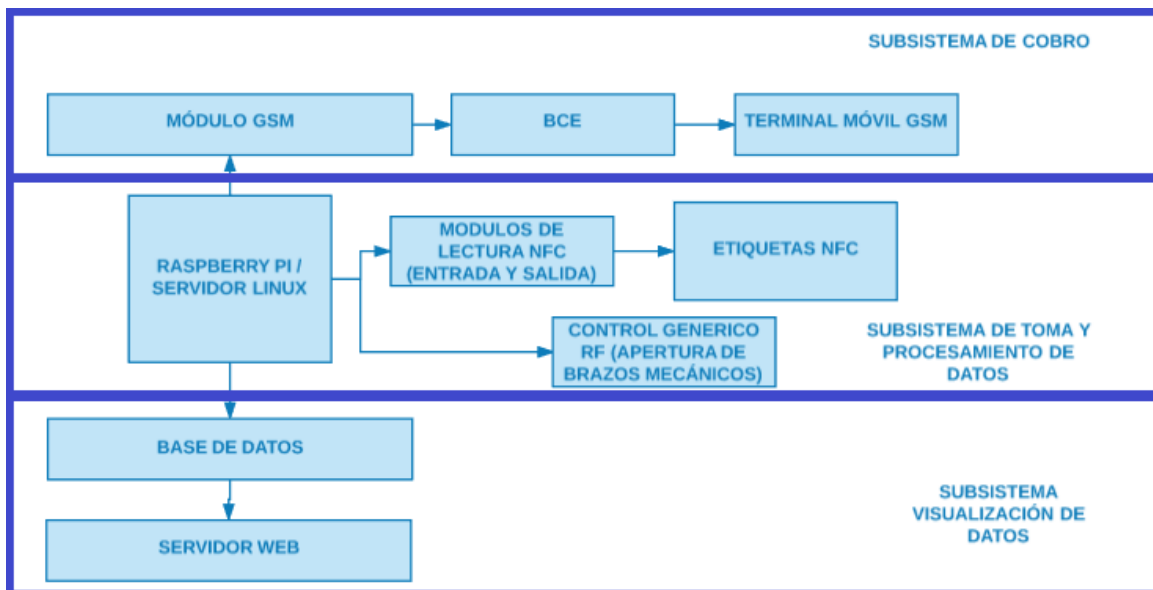


Figura 20. Esquema general de la estructura de cobro con dinero electrónico.

El subsistema de cobro está compuesto por los módulos Raspberry PI, Módulo GSM, Cuenta BCE y tecnologías indispensables para realizar el cobro como terminales móviles y plataforma de dinero electrónico del BCE.

El subsistema de toma y procesamiento de datos consta de un servidor Linux, una base de datos ejecutado sobre el microcomputador Raspberry PI, módulos de lectura de etiquetas NFC para entrada y salida del parqueadero. Finalmente consta de un control RF para apertura de los brazos mecánicos de ingreso y salida.

Por último el subsistema de visualización de datos está compuesto por un servidor Linux, sobre el cual se encontrará ejecutado la base de datos y un servicio WEB para el despliegue de datos.

2.2 Funcionamiento Del Prototipo

De conformidad a la arquitectura planteada, el microcomputador Raspberry PI estará encargado de almacenar el código principal para todos los módulos y proveer de los recursos necesarios para que los diferentes módulos interactúen entre sí.

Entre las principales funciones del prototipo se tienen:

Lectura de etiquetas NFC, las cuales contienen un identificador único el cual es requerido para almacenar y procesar la información en una base de datos. Además permitir la visualización de datos mediante un servicio web. Finalmente permitir la conexión a la red GSM para el envío de cobros mediante el protocolo USSD con la ayuda de terminales móviles y la apertura de los brazos mecánicos de entrada y salida.

2.3 Diseño De Subsistema De Cobro

El presente subsistema cumplirá con los requerimientos que permitirán la comunicación GSM con el BCE y enviará el cobro a la plataforma de BCE para debito del costo por tiempo tabulado, por lo que se encuentra estructurado con los siguientes módulos interrelacionados:

- Módulo GSM.

- Plataforma de cobro con dinero electrónico.
- Terminales móviles con tecnología GSM.

En la Figura 21 mostrada a continuación se puede observar gráficamente los componentes mencionados anteriormente.

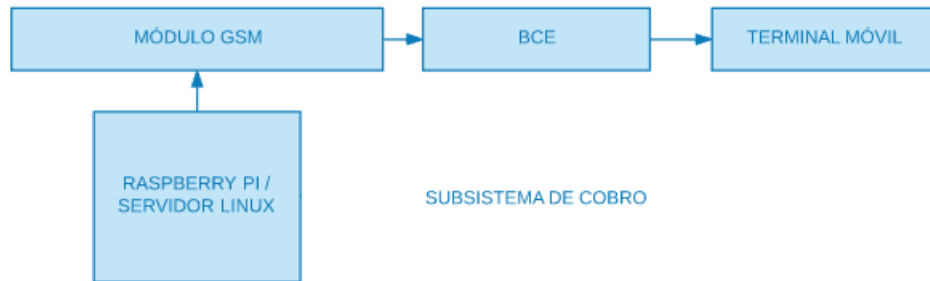


Figura 21. Diagrama de bloques del subsistema de cobro.

Este subsistema está encargado de efectuar el cobro con dinero electrónico. El proceso que realiza es el siguiente:

- Registro en la red GSM.
- Activa el modo texto para USSD en el módulo GSM.
- Enviar la trama USSD, armada con los datos tomados por el subsistema de toma y procesamiento de datos, mediante el módulo GSM conectado al Raspberry PI.
- Una vez realizado el proceso de cobro se escribe los resultados en la base de datos, los cuales servirán para activar el brazo mecánico mediante el control RF permitiendo la salida.

2.3.1 Tecnologías Aplicadas Al Subsistema De Cobro

El subsistema de cobro está compuesto por los siguientes elementos:

- Módulo GSM.
- Plataforma de cobro con dinero electrónico.

- Terminales móviles.

2.3.1.1 Módulo GSM

Es posible utilizar cualquier tipo de modem que posea tecnología GSM, dado que esta es requerida para comunicación USSD, necesaria para solicitar el envío del cobro. Para este prototipo se ha considerado utilizar el modem USB Huawei E3531.

2.3.1.2 Plataforma De Cobro Con Dinero Electrónico BCE

Actualmente la única entidad que ofrece servicios de dinero electrónico en el país es el Banco Central del Ecuador, por lo que este prototipo utilizará cuentas asociadas esta entidad monetaria.

2.3.1.3 Terminales Móviles

Es posible utilizar cualquier terminal móvil que posea tecnología GSM ya que esta permite comunicación USSD, necesaria para que el usuario pueda recibir el pago.

2.4 Diseño Subsistema De Toma Y Procesamiento De Datos

Este subtema cumplirá con los requerimientos de la toma de datos mediante escaneo NFC, tabulación del tiempo de estancia y permitirá la salida y entrada al parqueadero con la apertura de los brazos mecánicos, por lo que se encuentra estructurado con los siguientes módulos interrelacionados:

- Raspberry PI.
- Módulos NFC
- Etiquetas NFC.
- Base de datos.
- Control RF para apertura de brazos mecánicos.

En la Figura 22, es posible apreciar gráficamente los componentes por bloques mencionados anteriormente.

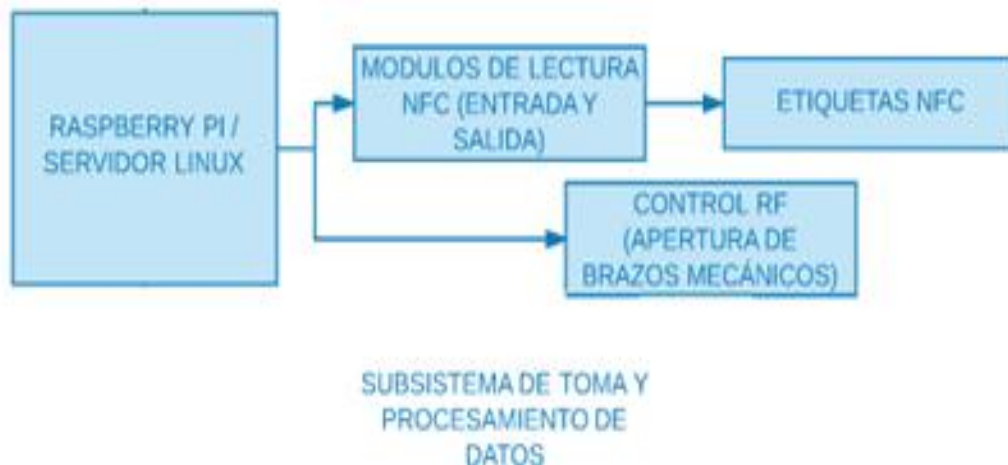


Figura 22. Diagrama de bloques del subsistema de toma y procesamiento de datos.

Este subsistema está dividido en tres bloques: el primero consta de la programación y configuración de los diferentes periféricos conectados al Raspberry PI, el segundo es para la toma de datos y por último el tercer bloque sirve para el procesamiento de los datos tomados y apertura de los brazos mecánicos.

2.4.1 Bloque De Toma De Datos

Este bloque está encargado de tomar datos, primero escanea mediante el módulo de entrada NFC la etiqueta que se encuentra asociada a un número de teléfono en la BDD, posterior almacena la información de la hora de ingreso asociado a la información tomada de la etiqueta NFC.

2.4.2 Bloque De Procesamiento De Datos

Este bloque es utilizado para la tabulación del valor a pagar por el uso del servicio. Primero escanea mediante el módulo de salida NFC la etiqueta que se encuentra asociada a un número y a una transacción previa al ingreso en la BDD, luego llena la información de la hora de salida. Finalmente calcula el valor a pagar por el uso del servicio.

2.4.3 Tecnologías Aplicadas Al Subsistema De Toma Y Procesamiento De Datos

El subsistema de toma y procesamiento de datos está compuesto por los siguientes elementos:

- Raspberry PI.
- Módulos NFC PN532v3.
- Etiquetas NFC.
- Base de datos.
- Control RF para apertura de brazos mecánicos.

2.4.3.1 Raspberry PI

Se ha considerado la plataforma Raspberry PI 3 cuyas características descritas en el subcapítulo 1.6.2, cumplen los siguientes requerimientos necesarios:

- Ser compatible con sistemas operativos Linux.
- Contar con puertos GPIO.
- Contar con tecnología *wifi*.
- Tener un tamaño reducido.
- Bajo costo.
- Permitir utilizar tareas automatizadas.

2.4.3.2 Módulos NFC PN532v3

Se ha considerado utilizar dos módulos PN532v3, por su fácil adquisición en el mercado y su reducido tamaño además de cumplir con los siguientes requerimientos:

- Ser compatible con Raspberry PI.
- Permitir lectura de etiquetas NFC.

2.4.3.3 Etiquetas NFC

Se ha considerado utilizar etiquetas NFC proporcionadas en la Universidad De Las Américas a los estudiantes como tarjeta de identificación interna, las cuales cuentan con tecnología NFC, dado que estas permiten la lectura necesaria para la toma de datos en el prototipo.

2.4.3.4 Base De Datos

Para la selección del software de base de datos se ha considerado los siguientes requerimientos:

- Compatibilidad con Raspberry PI.
- Permitir consumo de datos por servicio WEB.
- Utilice lenguaje SQL.
- Permita reducir costos en el prototipo.

De entre los diferentes software que cumplen los requerimientos y pueden ser ejecutados en el Raspberry PI se ha considerado utilizar la base de datos SQL MariaDB, dado que cumple con el objetivo de almacenamiento de datos y consulta a través de transacciones SQL, además de reducción de costos por ser software libre y encontrarse por defecto en los repositorios del sistema operativo “Raspbian” utilizado para el desarrollo del prototipo.

2.4.3.5 Control RF Para Apertura De Brazos Mecánicos

Es posible utilizar cualquier control RF que trabaje en la frecuencia 433MHz y que permita la clonación de código de otros controles RF, en este caso se utilizara un mando genérico.

2.5 Diseño De Subsistema De Visualización De Data

El subsistema de visualización de data cumplirá con el requerimiento de permitir la consulta del costo por el tiempo transcurrido, por lo que se ha estructurado con los siguientes bloques.

- Bloque servidor Linux
- Bloque de base de datos
- Bloque servicio WEB

En la Figura 23 es posible observar gráficamente los componentes de los bloques mencionados.

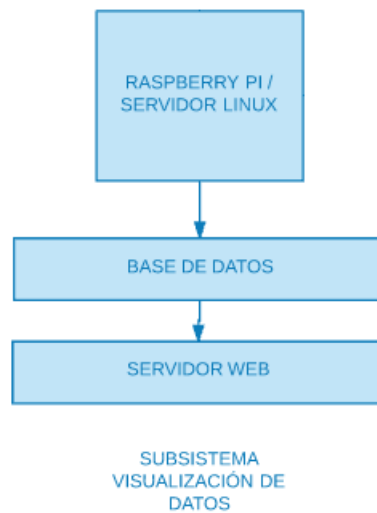


Figura 23. Subsistema de visualización de datos.

El subsistema de visualización de datos presentará el costo por el tiempo transcurrido en el parqueadero, además de permitir el pago emergente a la administración del establecimiento.

2.5.1 Tecnologías aplicadas al subsistema de visualización de datos.

2.5.2 Bloque De Base De Datos (Bdd)

Este bloque almacenara información del sistema y mediante consultas SQL brindará accesibilidad a datos específicos para luego ser mostrados en el bloque WEB.

Como se menciona fue definido anteriormente se ha seleccionado la base de datos SQL MariaDB, que para este bloque será utilizado gracias a las siguientes características:

- Compatible con Raspberry PI.
- Consumo de datos por servicio WEB.
- Trabaja con lenguaje SQL.
- Conectividad segura.
- Búsqueda e indexación de campos de texto.

2.5.3 Bloque Servicio Web

Este bloque es el encargado de obtener la información almacenada en la base de datos MariaDB, para mostrarla al usuario a través de un servicio WEB alojado en el Raspberry PI bajo el sistema operativo Linux Raspbian.

Una característica de este bloque es que se podrá acceder al servicio desde un dispositivo móvil conectado en la misma red del servidor Linux a través de una red wlan.

Para este prototipo se ha seleccionado utilizar apache dado que cumple con los siguientes requerimientos:

- Compatible con Raspberry PI.
- Permita reducción de costos.
- Maneje etiquetas HTML.
- Compatible con el lenguaje de programación PHP.
- Permita conectividad con la base de datos MariaDB.

2.5.3.1 Bloque Servidor Linux

Existen una variedad distribuciones GNU Linux que pueden ser instaladas en el Raspberry PI. Para el desarrollo de este prototipo se ha considerado utilizar la distribución Raspbian, como sistema operativo, ya que este es de distribución

libre; además es desarrollado específicamente para el Raspberry PI. Este SO permite cumplir con los requerimientos que se mencionan a continuación:

- Toma y almacenamiento de datos desde el subsistema de toma y procesamiento de datos.
- Envío de tramas USSD desde el subsistema de cobro.
- Extracción de información desde la base de datos para presentarlos en una interfaz WEB.

2.6 Presentación De Tecnología Seleccionada

A continuación se presentara la tecnología que se utilizará en el siguiente capítulo para implementar el prototipo presente:

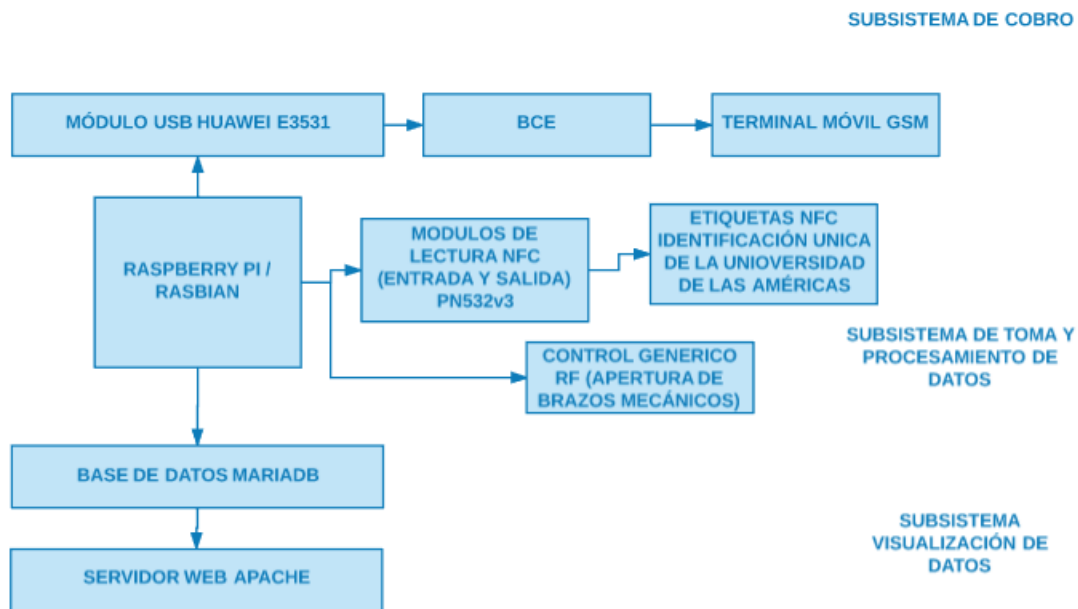


Figura 24. Presentación de tecnología seleccionada.

3. IMPLEMENTACIÓN Y PRUEBAS DEL PROTOTIPO

En el presente capítulo se presentará la configuración de los bloques de cobro, toma y procesamiento de datos y visualización de datos, así como la función que

cumplen en el sistema. Posteriormente se presentara un flujo de pruebas para validar el correcto funcionamiento del prototipo. Finalmente se denotara los resultados de las pruebas realizadas.

3.1 Implementación De Los Subsistemas

3.1.1 Implementación Del Subsistema De Toma Y Procesamiento De Datos

El subsistema de toma y procesamiento de datos consiste en un servidor Linux ejecutado sobre el microcomputador Raspberry PI, el cual contiene la lógica necesaria para la toma, almacenaje y procesamiento de data mediante los diferentes módulos mencionados en los tres subsistemas que componen el prototipo. Además de realizar la lógica para permitir la apertura de los brazos mecánicos mediante el control RF.

En este subcapítulo se mencionara la lógica necesaria desarrollada en Python ejecutada sobre el Raspberry PI la cual cumple con las necesidades que se requieren para cumplir con el cobro mediante dinero electrónico, posterior describirá la integración de los módulos NFC, para luego dar paso al funcionamiento de la base de datos y finalmente la configuración del control RF para apertura de brazos mecánicos.

3.1.1.1 Integración Y Configuración De Módulos Al Raspberry PI

Para que los diferentes módulos puedan interactuar entre sí, es necesario un programa master que utilice los recursos de los periféricos conectados al Raspberry PI.

El Raspberry PI manejará las funciones de apertura de brazo mecánico de ingreso mediante el control RF, toma y almacenamiento de información en la base de datos, la cual será adquirida desde las etiquetas NFC, para posteriormente realizar el cálculo del monto a pagar, necesario para generar la trama USSD que será enviada mediante el subsistema de cobro para que el usuario realice el pago

mediante dinero electrónico, finalmente validará el pago y realizará la apertura del brazo mecánico que permitirá la salida del usuario del parqueadero.

Para la integración y configuración de los módulos utilizados en este prototipo se requiere instalar librerías sobre el SO raspbian, además de la creación de scripts que hagan uso de los recursos conectados al Raspberry.

3.1.1.1.1 Módulos NFC PN532v3

Para que Raspberry Pi pueda hacer uso de los módulos PN532v3 estos deben estar conectados mediante I2C (circuito integrado) a los puertos del Raspberry como se muestra en la Figura 25.

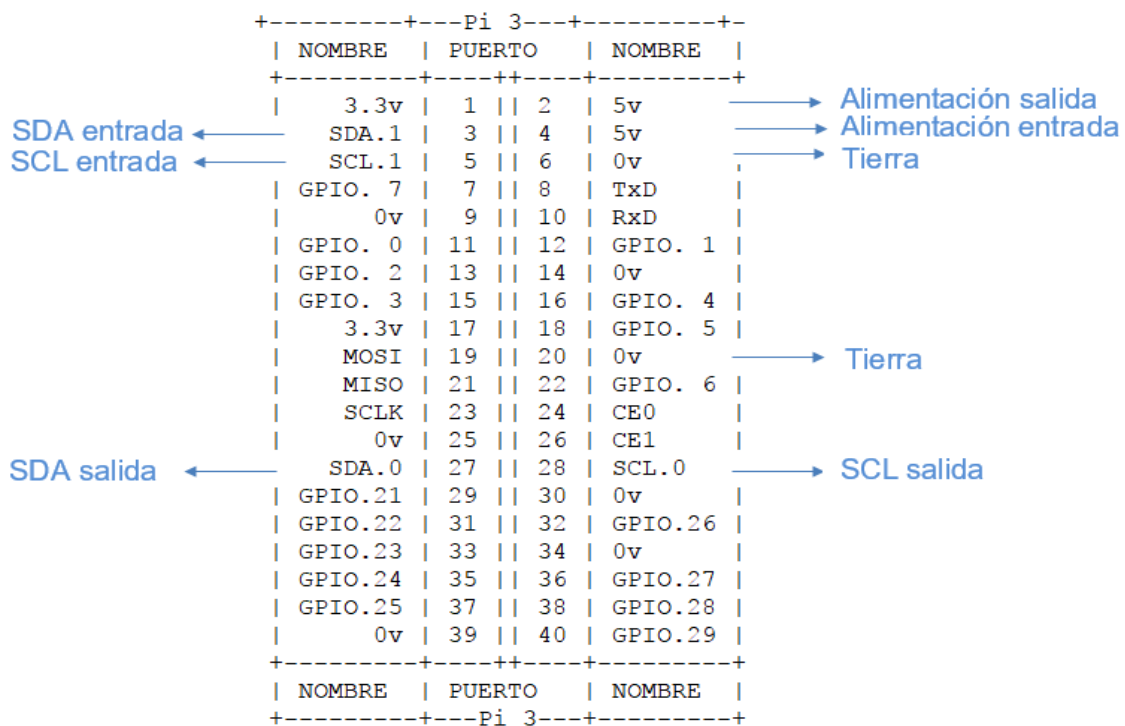


Figura 25. Conexión PN532v3 al Raspberry Pi.

Para hacer uso de los módulos NFC conectados a los puertos GPIO del Raspberry Pi es necesario que se active la opción de uso de puertos I2C, instalar la librería *python-smbus* e *i2c-tools*.

Para realizar el proceso de lectura para entrada y salida se utiliza dos scripts en python descritos a continuación:

- `in_parking.py` permite realizar la lectura de las etiquetas NFC y con la ayuda de clases del archivo `sql.py` realizar la escritura de datos en MariaDB como se muestra en el flujo grama de la Figura 26.

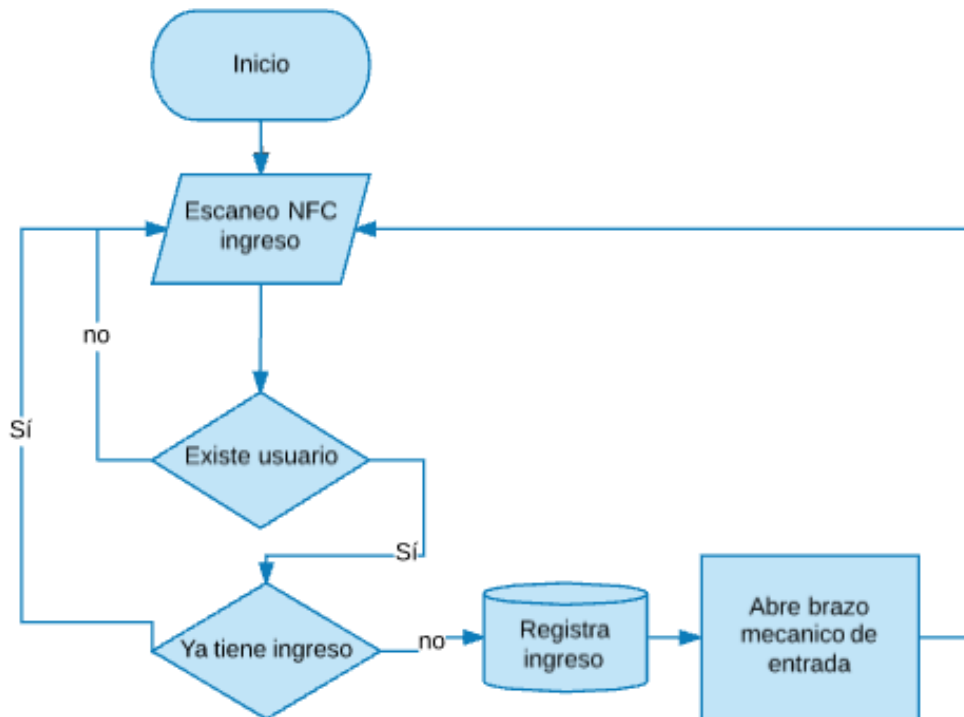


Figura 26. Flujo de entrada script `in_parking.py`.

- `exit_parking.py` permite realizar la lectura de etiquetas NFC para la salida y con la ayuda del archivo `sql.py` realiza lectura y escritura de datos para cumplir con la función del cobro y apertura del brazo mecánico ayudándose del script `gsm_transactions.py` y para concluir realiza la apertura del brazo mecánico el flujo completo es mostrado en la Figura 27.

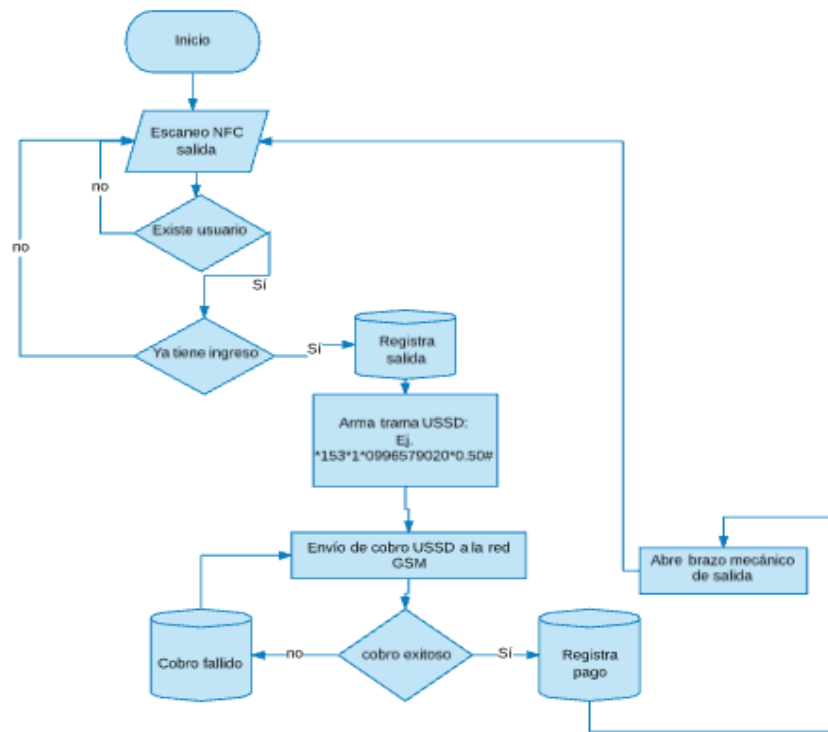


Figura 27. Flujo de salida script exit_parking.py

3.1.1.1.2 Módulo GSM Huawei E3531

El módulo Huawei E3531 debe estar conectado al Raspberry a cualquiera de los puertos USB como se muestra a continuación en la Figura 28.



Figura 28. Acoplamiento módem GSM al Raspberry Pi.

Para hacer uso del módulo GSM conectado es necesario la instalación de la librería python-rpi.gpio.

El script gsm_transactions.py permite realizar el envío de la trama USSD del cobro mediante el módulo GSM, posteriormente valida que este sea exitoso como se observa en la Figura 29 a continuación.

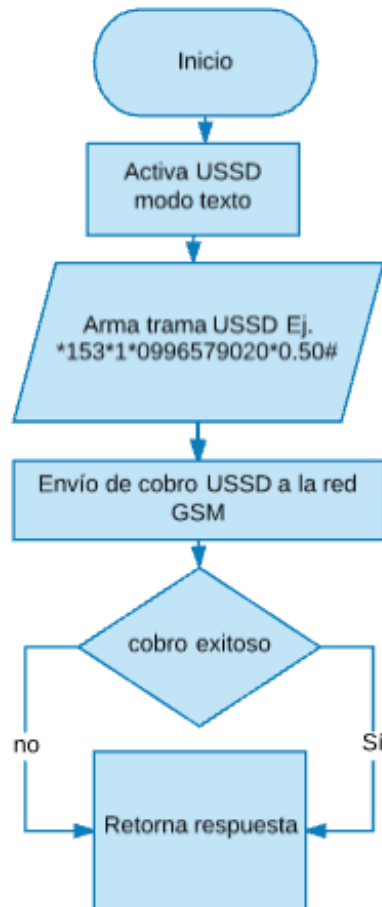


Figura 29. Envió de cobro a la red GSM.

3.1.1.1.3 Base De Datos MariaDB

Previa instalación de MariaDB sobre el SO raspbian, es necesaria la creación de una base de datos llamada COBRO, con un esquema de tablas relacionales, las

cuales contendrán la estructura para que el prototipo pueda extraer e ingresar datos de manera efectiva. Esta estructura se encuentra conformada por 4 tablas:

- USUARIO
- ESTADO_PAGO
- TIEMPO_USADO
- TRANSACCION

Las tablas se encuentran relacionadas entre sí como se puede observar en la Figura 30.

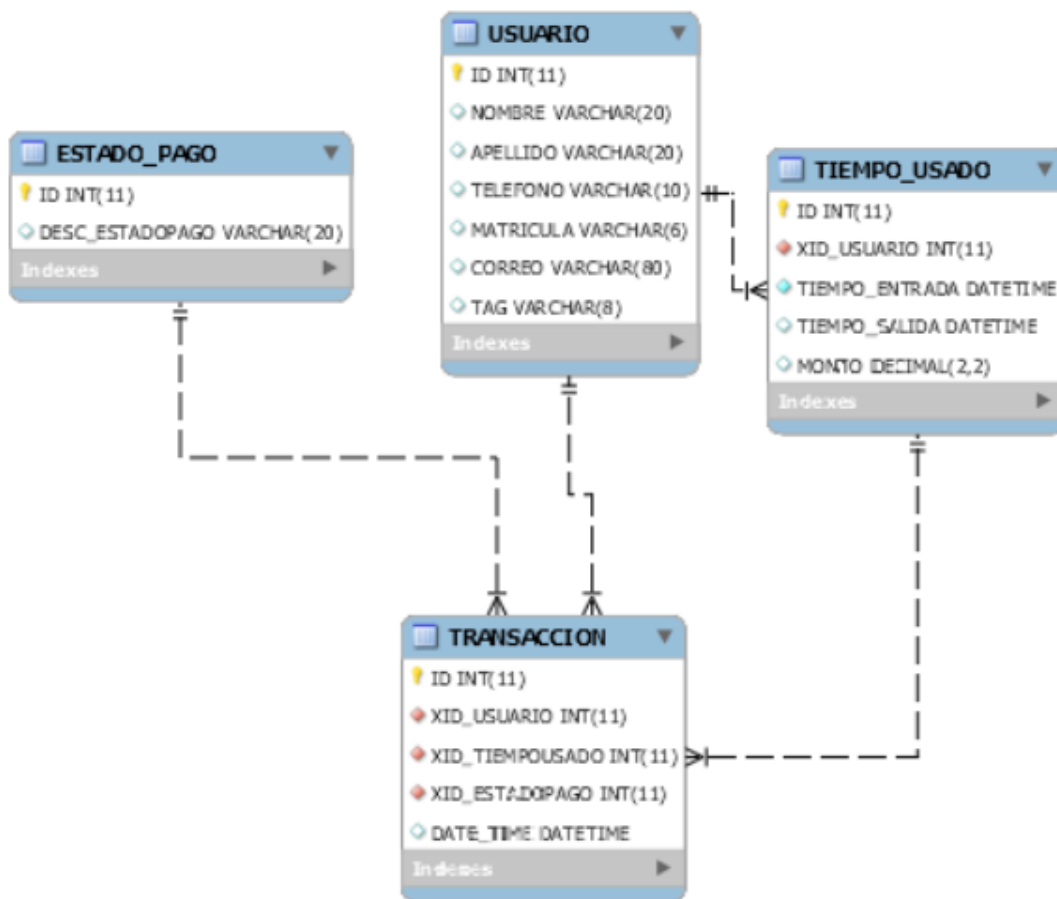


Figura 30. Esquema relacional de tablas.

Para realizar transacciones en este esquema de datos es necesario la utilizar el archivo python sql.py, el cual contiene clases con sentencias definidas para escritura y lectura en la base de datos.

3.1.1.1.4 Configuración Control RF

La función principal de este control es realizar la apertura de los brazos mecánicos tanto de entrada como de salida del parqueadero.

El control requiere previa configuración para su funcionamiento, a continuación se menciona el modo de clonación de códigos de mandos RF.

Métodos de operación

1. Eliminar el código Presione la tecla Bloquear y desbloquee la tecla al mismo tiempo, dos segundos más tarde, el LED parpadea 3 veces, seguido de un parpadeo rápido continuo que indica que el código de control almacenado fue eliminado.
2. Duplicación de código, para que la duplicación sea posible el mando original debe estar lo más cerca posible del mando genérico, una vez juntados ambos controles se debe presionar los botones de bloqueo de los dos mandos a distancia al mismo tiempo; entonces el led parpadea 3 veces después de 3 segundos, después de un parpadeo rápido. Indica que el control original del código de la llave de bloqueo se ha copiado en el control remoto duplicado.

El mando a distancia es accionado en dos instancias: una cuando el usuario acerca la etiqueta NFC al lector para permitir el ingreso al parqueadero y la otra, luego que se valide que el cobro es exitoso o a su vez se a realizado el pago manual de emergencia.

Para permitir la interacción del mando RF con el raspberry Pi es necesario un circuito acoplador, mostrado en la Figura 31, que integre el mando con el Raspberry Pi.

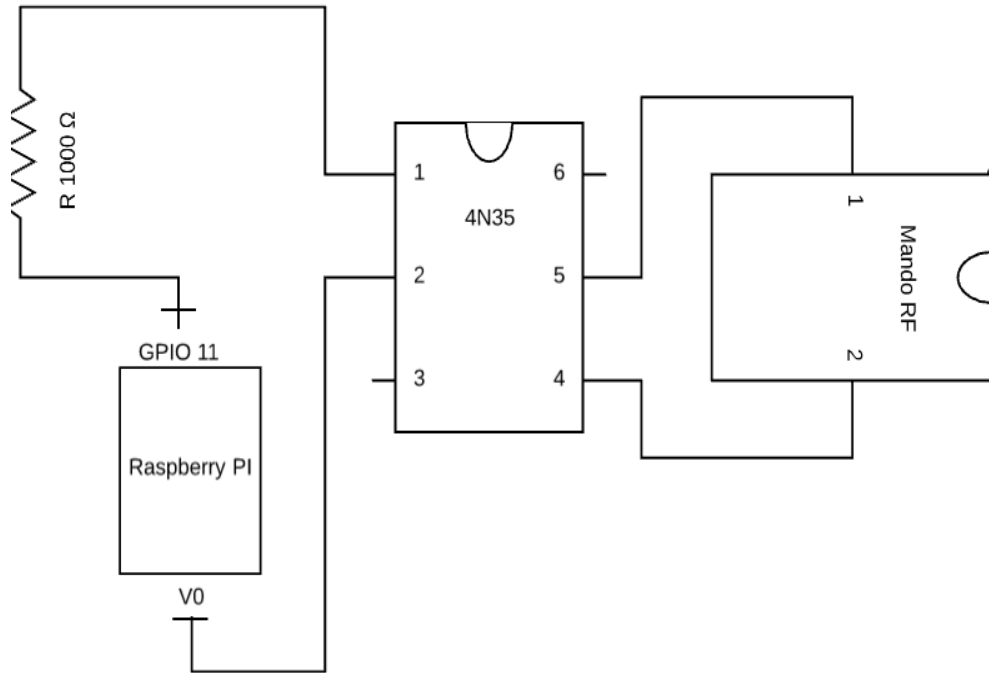


Figura 31. Circuito acoplador con Raspberry Pi.

La función de este acople es accionar el mando RF, cuando el raspberry PI envía voltaje al pin 1 del integrado 4N35 cierra el circuito del mando a distancia accionándolo, esto realiza la apertura del brazo mecánico.

Para que el mando pueda ser accionado es necesario incluir la clase `rf_control.py`, que permite enviar voltaje al circuito para que el mando RF sea accionado.

3.1.1.2 Funcionamiento Del Sistema De Cobro Con Dinero Electrónico

En la Figura 32 se aprecia el flujo completo del programa ejecutado sobre la plataforma Raspberry PI.

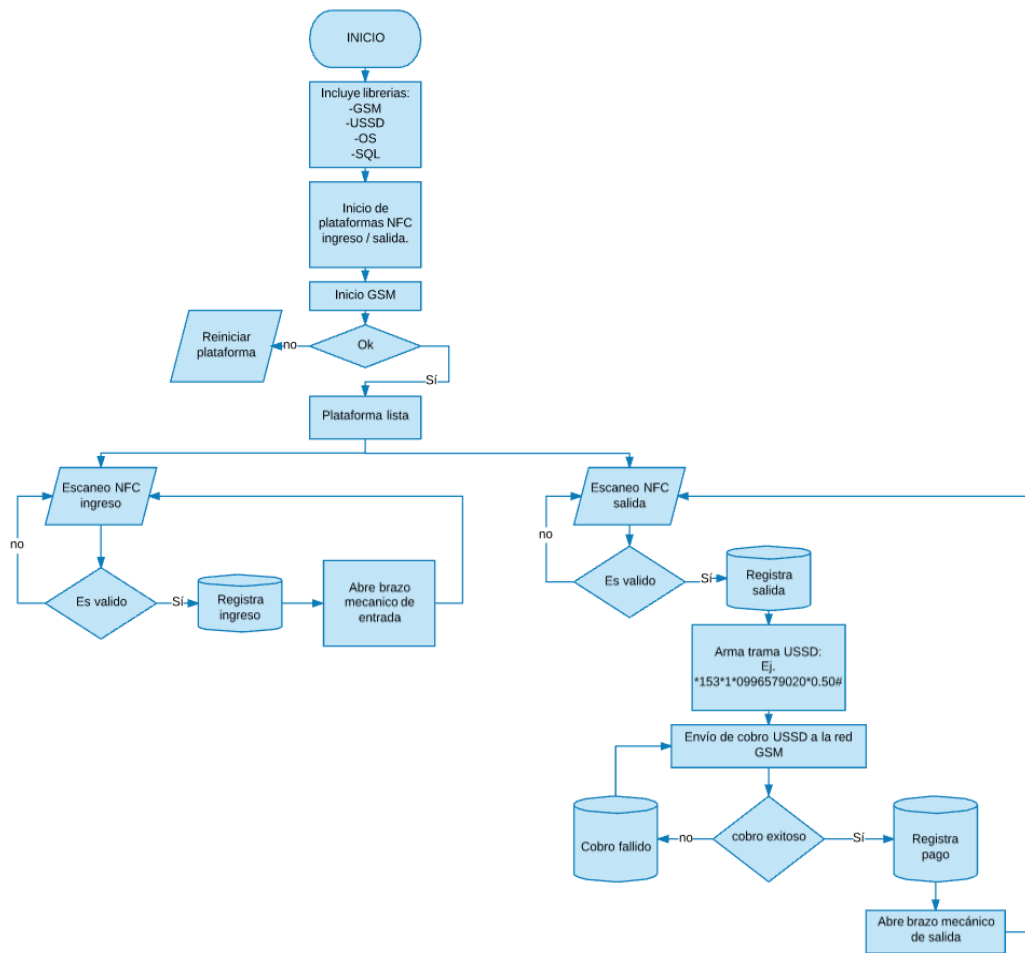


Figura 32. Flujo completo del sistema de cobro con dinero electrónico.

El flujo inicia con incluir las librerías necesarias para la correcta ejecución del programa, seguido de la inicialización de los módulos GSM y NFC, donde se realiza una validación del correcto funcionamiento de estos periféricos, una vez lista la plataforma se inicia la fase de escaneo de etiquetas. En la fase del ingreso se valida que la etiqueta NFC se encuentre registrada en el sistema y que no tenga un previo ingreso, si se cumple esta validación se registra el ingreso en la base de datos y se realiza la apertura del brazo mecánico. La fase de salida valida que exista un ingreso previo, que la etiqueta se encuentre registrada en el sistema y que no exista un pago emergente realizado, si cumple estas validaciones se registra la salida y se toma el número de móvil y el monto a pagar para

posteriormente enviar el cobro mediante el protocolo USSD y realizar un escaneo de la red GSM para validar que el cobro sea efectuado. Finalmente se realiza la apertura del brazo mecánico de salida.

3.1.2 Implementación Del Subsistema De Cobro

El subsistema de cobro está constituido con tecnología de hardware y software que permitirán cumplir las funciones establecidas en el subcapítulo 2.4 donde se mencionara la configuración del módulo GSM y la creación de cuentas de dinero electrónico en el BCE.

3.1.2.1.1 Módulo GSM Huawei E3531

Este dispositivo es el encargado de realizar la comunicación GSM y a su vez enviar mensajes USSD que permiten realizar la función de cobro entre la cuenta de BCE en el prototipo y cliente.

El presente módulo requiere de un chip de telefonía GSM que puede ser de cualquier operadora. Este chip GSM debe estar ligado a una cuenta de dinero electrónico con RUC para poder realizar cobros.

Un punto importante a considerar es armar la trama USSD, que es el mensaje que se enviara mediante la red GSM hacia el BCE que a su vez solicitara el cobro al cliente como se observa en la Figura 33.

La sintaxis de la trama USSD es la siguiente: “*153*1*numerocliente*valorporpagar#”, la cual requiere del prefijo *153* el que invoca el servicio del dinero electrónico del BCE, seguido de la opción 1 el cual permite enviar el cobro, posteriormente es necesario el número de teléfono de la persona a la que se le enviara el cobro, este número es tomado a partir de la información almacenada en la base de datos que se encuentra relacionada a la etiqueta NFC presentada en los lectores NFC. Finalmente el último parámetro

necesario para la trama de cobro USSD es tomada de la tabulación del tiempo de estancia en el parqueadero igualmente relacionada a la etiqueta NFC.

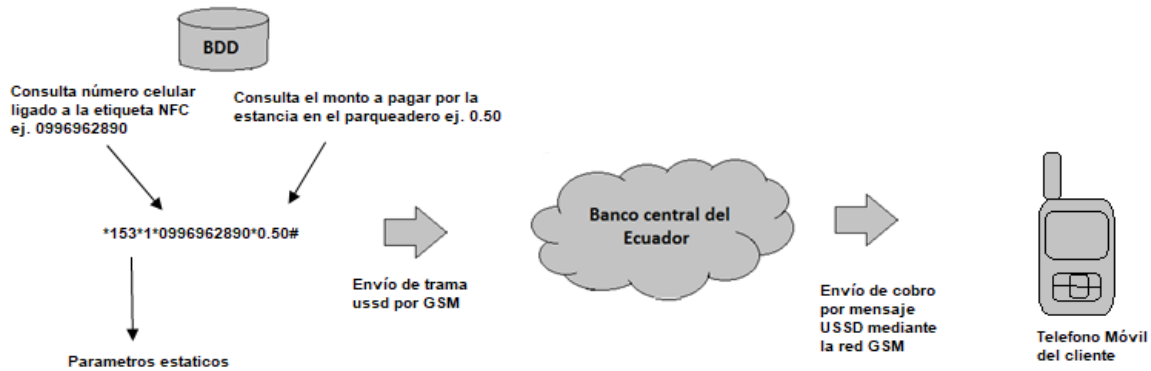


Figura 33. Flujo de envío de cobro por la red GSM.

La trama USSD es construida tras consultar el número móvil y monto a pagar asociados a la etiqueta NFC en la base de datos. Posteriormente es enviada a la red GSM mediante la clase o porción de código python indicado en la siguiente Figura 34.

```

def gsm_cobro(num, monto):
    num = str(num)
    monto = str(monto)
    ser.write("AT^USSDMODE=0\r")
    time.sleep(2)
    trama = "AT+CUUSD=1, \"*153*2*"+num+"*"+monto+"#\r"
    ser.write(trama)
    print trama
    status=wait_response("exito", "progreso.", "ERR", "919")
    print "Retorno estado: "+str(status)
    return status
  
```

Figura 34. Envío mensaje USSD

El comando `ser.write` permite el envío de la trama USSD a la red GSM para ejecutar el cobro, posteriormente se lee la red GSM en busca de respuestas

USSD con la clase `wait_response` que acepta cuatro parámetros de tipo cadena de caracteres los cuales se utilizarán para validar si es una respuesta positiva o negativa. En este caso se ha proporcionado la palabra “exito” que será la palabra esperada para confirmar que el cobro fue realizado, “progreso” que quiere decir que el cobro ha sido enviado y finalmente “ERR” y “919” que definirán que el cobro no fue realizado. En el código mostrado en la Figura 35 se define el flujo de estas respuestas.

```
def wait_response(resp, resp2, resp3, err):
    reply = ser.read(ser.inWaiting())
    reply = str(reply)
    resp = str(resp)
    resp2 = str(resp2)
    resp3 = str(resp3)
    err = str(err)
    tout = 0
    print "Esperando:" + resp + ", "+resp2+", "+resp3+", "+err

    while (reply != resp):
        if (tout <= 800000):
            tout = tout + 1
            reply = ser.read(ser.inWaiting())
            if re.search(resp3 , reply):
                print "Se encontro en la trama: "+reply
                resp = reply
                return "0"
            if re.search(resp2 , reply):
                print "Se encontro la trama:"+reply
            if re.search(resp , reply):
                print "Se encontro la trama"+reply
                resp = reply
                return "1"
            if re.search(err , reply):
                print "Se encontro la trama"+reply
                resp = reply
                return "0"
        else:
            ser.write("AT+CUSD=2")
            resp=reply
            print "Time out bye!"
            return "0"
```

Figura 35. Lectura de respuesta USSD en la red GSM.

La clase recibe 4 parámetros con los cuales se realizara condicionales en busca de respuestas escaneadas con el comando *ser.read*, estas condicionales devuelven 1 o 0 dependiendo si la respuesta esperada fue positiva o negativa. Se muestra el diagrama de flujo que cumple esta clase en la Figura 36.

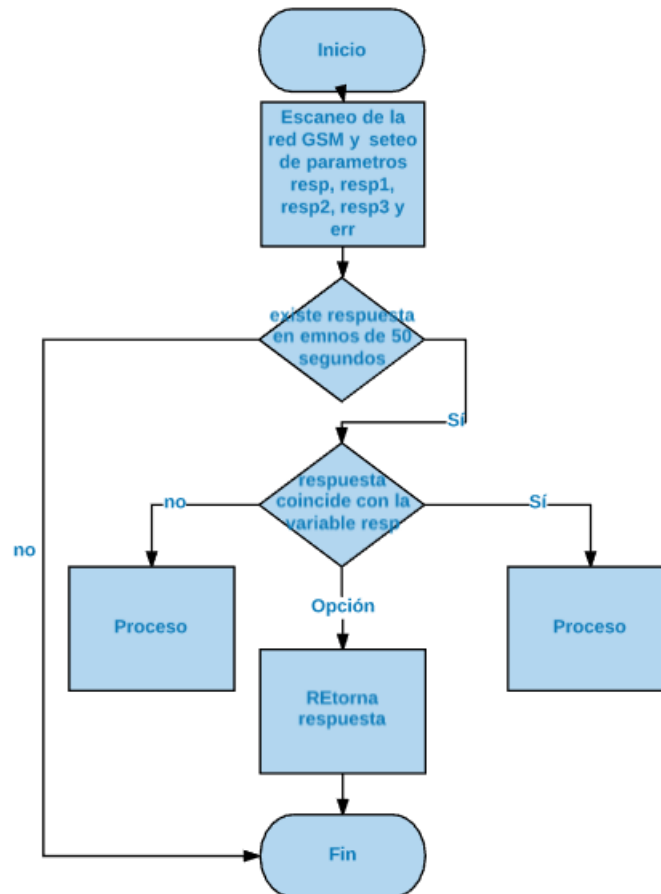


Figura 36. Diagrama de flujo escaneo USSD de red GSM.

3.1.2.1.2 Configuración De Una Cuenta En BCE

Para realizar transferencias con dinero electrónico es necesario tener una cuenta activa en el BCE, a continuación, se describirá los pasos que son necesarios para la creación de cuentas de dinero electrónico en el BCE.

Para la apertura de una cuenta de dinero electrónico para persona natural la plataforma solicita la siguiente información:

- Aceptar la creación de la cuenta.
- Aceptar los términos y condiciones de uso como se puede observar en la siguiente Figura 37.

Figura 37. Descripción grafica del punto 1 y 2.

- Seleccionar persona natural con RUC.
- Ingresar RUC mostrado a continuación en la Figura 38.

Figura 38. Descripción grafica punto 3 y 4.

- Confirmar nombres y apellidos como se muestra en la Figura 39.

Figura 39. Descripción grafica punto 5.

- Se debe responder preguntas de seguridad (fecha de nacimiento, estado civil, lugar de nacimiento) las preguntas son mostradas en la Figura 40 a continuación:

| | | |
|--|---|--|
| <p>Ingresa tu fecha de nacimiento (con el formato ddmmAAAA)</p> <input type="text"/> <p>CANCELAR ENVIAR</p> | <p>Selecciona tu lugar de nacimiento:</p> <ol style="list-style-type: none"> 1.-CHIMBO 2.-LOJA 3.-QUITO 4.-ORO <input type="text"/> <p>CANCELAR ENVIAR</p> | <p>Selecciona tu estado civil:</p> <ol style="list-style-type: none"> 1.-SOLTERO 2.-CASADO 3.-DIVORCIADO 4.-VIUDO 5.-UNIDO 6.-EN UNION DE HECHO <input type="text"/> <p>CANCELAR ENVIAR</p> |
|--|---|--|

Figura 40. Graficas de preguntas punto 6.

- Un mensaje de texto llega confirmando la apertura de la cuenta y clave temporal de seguridad es recibido en el móvil del contratante como se muestra en la Figura 41.

| | |
|---|--|
| <p>Se enviara un mensaje de confirmación. Su clave temporal es 6671</p> | <p>Gracias por registrarse, tu clave temporal es 6671. Para empezar a utilizar Efectivo desde mi celular marca *153#</p> |
|---|--|

Figura 41. Mensajes de texto de confirmación.

- Se marca nuevamente *153# opción 1, opción 4 cambio de clave como se muestra en la siguiente Figura 42.

Mi cuenta

- 1.-Cobros
- 2.-Consulta de Saldo
- 3.-Consulta ultimas transacciones
- 4.-Cambio de clave
- 5.-Gestion de monederos

CANCELAR ENVIAR

Figura 42. Grafica punto 8.

En el caso de apertura de una cuenta de Dinero Electrónico con cedula se sigue el mismo flujo con el único cambio al seleccionar persona natural en el punto 4.

El funcionamiento de persona natural con RUC y persona natural se diferencian en que la cuenta natural con RUC permite realizar cobros y pagos, mientras que la cuenta de la persona natural solo permite realizar pagos, a continuación en la Figura 43, se puede diferenciar el menú desplegado para cada uno.

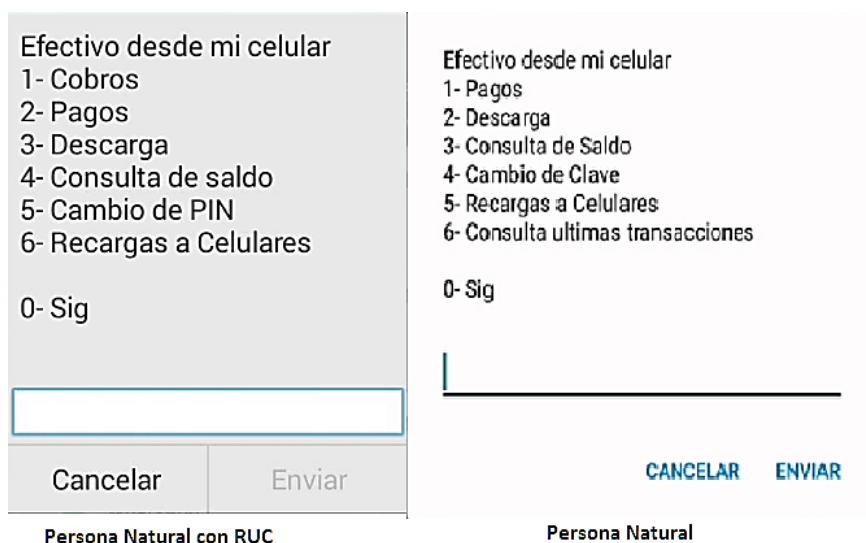


Figura 43. Diferencia entre menú Persona Natural y Persona Natural con RUC.

3.1.3 Implementación Del Subsistema De Visualización De Data

El presente subsistema está constituido por un servicio web que permite consultas a la base de datos MariaDB en donde es necesario la conexión a la misma red local que el servidor para acceder a los servicios web.

Este servicio web permite la consulta del monto en dólares por la estancia transcurrida en el parqueadero. Para acceder a este servicio se requiere el ingreso del número de matrícula relacionada a la etiqueta NFC en la base de datos como se muestra en la Figura 44.



Figura 44. Ingreso de matrícula en interface web de consulta.

Ingresado el número de matrícula se procederá a presentar el monto por pagar o mostrara el texto “no tiene saldo por pagar” como se muestra en la Figura 45 a continuación.

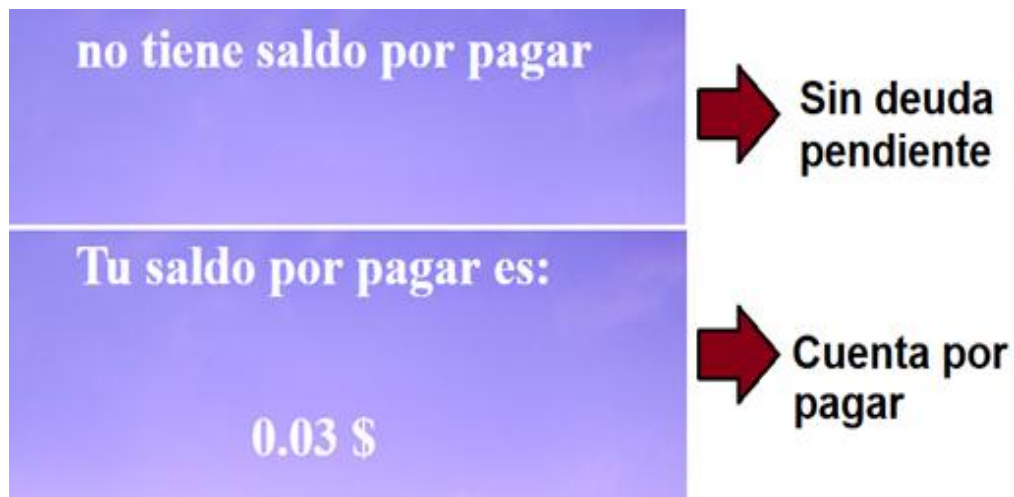


Figura 45. Consulta de monto por pagar.

3.1.3.1 Pago De Emergencia

Esta interface permite cancelar de manera manual el valor por uso del parqueadero, en caso de cualquier escenario presentado que no permita efectuar el pago con la utilización de dinero electrónico. A este módulo tiene acceso únicamente una persona encargada de la administración del parqueadero que contará con un usuario y contraseña a diferencia de la interface de solo consulta que no requiere de estos campos, como se muestra en la Figura 46 siguiente:



The image shows a web-based login interface. At the top, the title "Login de Usuarios" is displayed in a large, stylized font. Below the title, there are two input fields: the first is labeled "Nombre Usuario:" and the second is labeled "Password:". Below these fields is a button labeled "LOGIN". At the bottom of the interface, the URL "www.udla.edu.ec" is displayed. The background of the interface is a photograph of a multi-story building at dusk or night, with some windows illuminated.

Figura 46. Acceso por usuario y contraseña.

dentro del servicio de administrador web se presenta una pantalla la cual se puede observar en la Figura 47 donde solicita el ingreso de matrícula, posteriormente muestra el valor a pagar y la opción de realizar el pago manual.



Figura 47. Interface de pago manual.

Concluido el pago manual se mostrara un mensaje el cual se puede observar en la Figura 48. Cuando el usuario presente la etiqueta NFC en el lector de salida el sistema reconoce que ya existe un pago manual realizado y permite la salida del parqueadero.

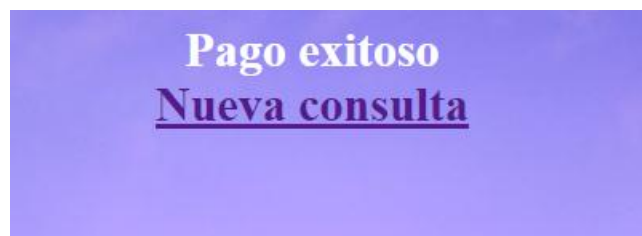


Figura 48. Pago manual exitoso.

3.2 Pruebas Y Resultados

En este subcapítulo se presenta las pruebas efectuadas con el prototipo implementado, los resultados obtenidos de las mismas y finalmente los costos de referencia del sistema prototipo.

3.2.1 Componentes De *Hardware* Y *Software* Para Pruebas De Funcionamiento

Los componentes de hardware y software empleados para la realización de las pruebas de funcionamiento se encuentran mencionados en la Tabla 4.

Tabla 4.

Hardware y software empleado en las pruebas del prototipo.

| SUBSISTEMA DE COBRO | |
|--|---------------------|
| EQUIPO / SOFTWARE | MODELO |
| Módulo GSM | Huawei E3531 |
| Terminal GSM 1 | Iphone 5s |
| SUBSISTEMA DE TOMA Y PROCESAMIENTO DE DATOS | |
| Raspberry Pi | Raspberri Pi 3 |
| Módulos NFC | PN532v3 |
| MariaDB | MariaDB 10.1.23 |
| Control RF | Generico |
| SUBSISTEMA DE VISUALIZACIÓN DE DATA | |
| Servidor linux | Raspbian Lite |
| PHP | PHP 7.0.19-1 |
| APACHE | APACHE 2 (raspbian) |
| MariaDB | MariaDB 10.1.23 |

Para la de pruebas es posible utilizar cualquier terminal celular que cuente con tecnología GSM ya que posee la capacidad de utilizar mensajes USSD necesario para que se realice el cobro, sin embargo, se realizará las pruebas el terminal mostrado en la Figura 49. Terminal GSM para pruebas..



Figura 49. Terminal GSM para pruebas.

Las características de estos terminales se muestran a continuación en la Tabla 5.

Características del terminal GSM usado en pruebas.

Tabla 5.

Características del terminal GSM usado en pruebas.

| Característica | IPHONE 5S |
|--------------------------|-----------------------------------|
| Memoria y almacenamiento | 1 GB RAM Memoria interna 16 GB |
| Capacidad de SIM | Nano SIM |
| Sistema Operativo | IOS 11.1.2 (15B202) |
| Procesador | Procesador Apple A7 |
| Batería | Batería 1560 mAh |

| | |
|------------------|--|
| Peso | 112 gramos |
| Dimensiones | 123.8 x 58.6 x 7.6 mm |
| Pantalla | Pantalla 4", 640 x 1136 <i>pixels</i> |
| Cámara principal | 8 MP, 3264x2448 <i>pixels</i> |
| Redes | GSM, GPRS, UMTS, HSPA, LTE. |

3.2.2 Escenario De Pruebas

Las pruebas se han realizado en un ambiente controlado con las siguientes características del sistema:

- El programa master debe estar ejecutándose.
- Los números de pruebas deben constar con el servicio de dinero electrónico del BCE.
- El área geográfica de pruebas debe tener cobertura GSM.
- Las etiquetas NFC deben encontrarse registradas en el sistema.
- El Raspberry PI debe encontrarse conectado a una red al igual que el móvil de consulta.
- Servicios Apache, MariaDB deben estar ejecutándose.

3.2.3 Pruebas De Funcionalidad

En este segmento se determinará el correcto funcionamiento del prototipo.

Para esta fase se ha definido las siguientes fases:

- Fase de ingreso al parqueadero.

- Fase de consulta y procesamiento de datos.
- Fase de envío y resultado de cobro.
- Fase de pago emergente.

3.2.3.1 Fase De Ingreso Al Parqueadero

Una vez iniciado el sistema el prototipo se encuentra listo para la fase de escaneo, en el cual el módulo NFC PN532v3 realiza el escaneo de etiquetas como se observa en la Figura 50, donde compara la información recibida con información almacenada en la base de datos en busca de concordancia de usuario. Una vez que encuentra la concordancia de usuario crea un registro nuevo en la tabla TIEMPO_USADO, con el ID del usuario encontrado, sin llenar los parámetros de TIEMPO_SALIDA y MONTO, también crea un registro en la tabla TRANSACCION con el campo XID_ESTADOPAGO 1 o nuevo en la base de datos tal como se evidencia en la Figura 50.

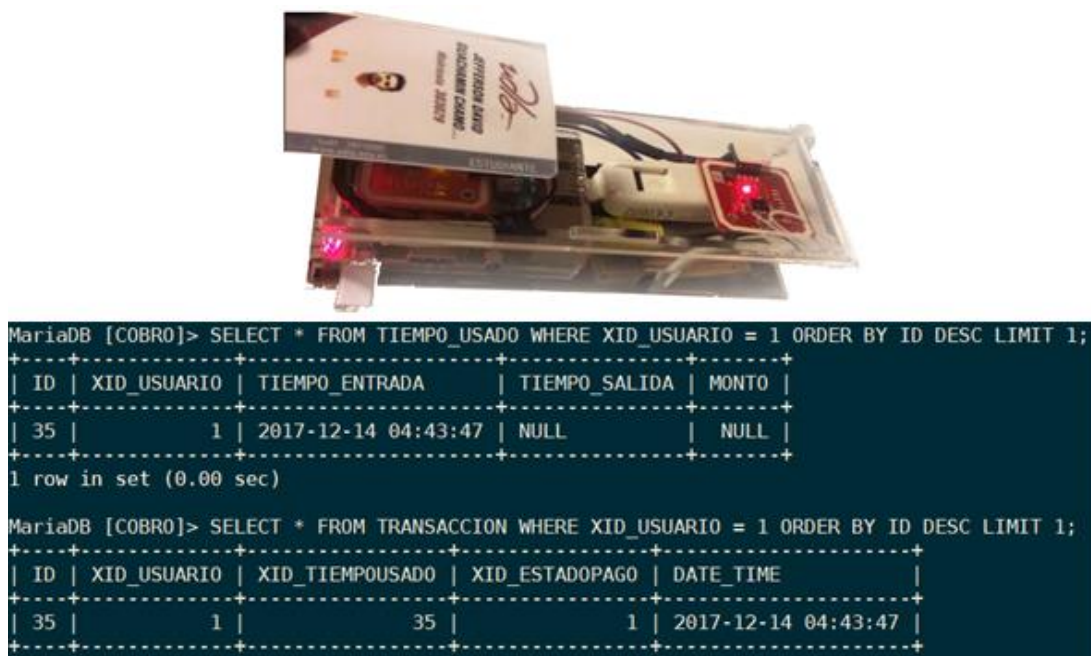


Figura 50. Lectura de etiqueta NFC y registro de ingreso en la BDD.

Posteriormente el Raspberry Pi realiza el envío de voltaje al integrado 4N35 para que cierre el circuito y permita que el mando se accione enviando la señal de apertura del brazo mecánico permitiendo el ingreso al parqueadero al usuario.

3.2.3.2 Fase De Consulta Y Procesamiento De Datos

En esta fase el usuario consulta el valor actual a pagar por el tiempo transcurrido desde el ingreso al parqueadero como se observa en la Figura 51. Se recalca que este no será el valor final, dado que este se hace definitivo una vez que se intente la salida del establecimiento presentando la etiqueta NFC al lector de salida.

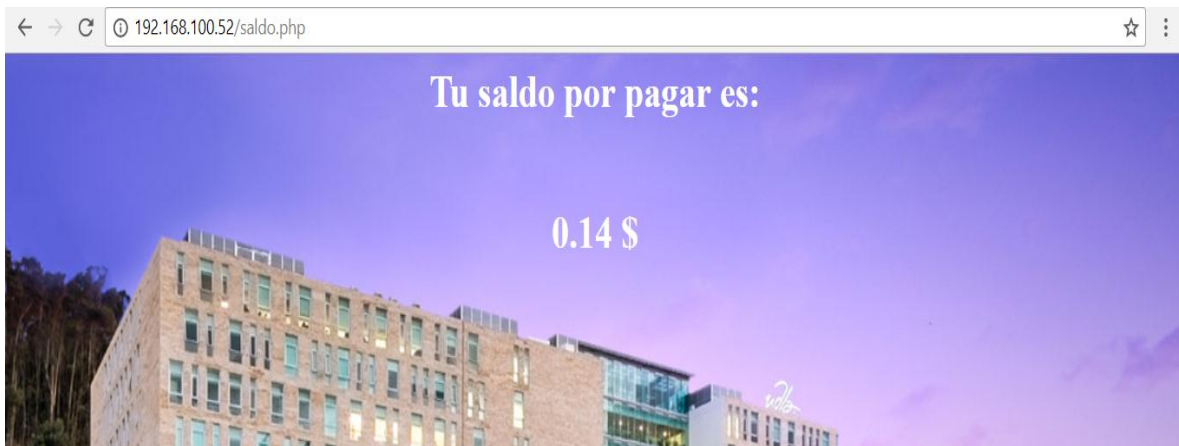


Figura 51. Consulta de monto por pagar mediante la interface WEB.

Luego de presentar la etiqueta NFC en el módulo de salida, esta es registrada en la base de datos en la tabla TIEMPO_USADO con la hora de la lectura de la etiqueta NFC, posteriormente el monto es calculado y escrito en la base de datos como se evidencia en la Figura 52.

```
MariaDB [COBRO]> SELECT * FROM TIEMPO_USADO WHERE XID_USUARIO = 1 ORDER BY ID DESC LIMIT 1;
+-----+-----+-----+-----+-----+
| ID | XID_USUARIO | TIEMPO_ENTRADA | TIEMPO_SALIDA | MONTO |
+-----+-----+-----+-----+-----+
| 35 | 1 | 2017-12-14 04:43:47 | 2017-12-14 05:21:06 | 0.31 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Figura 52. Registro de salida en la base de datos.

Posterior al registrado del monto se valida que el XID_ESTADOPAGO en la tabla TRANSACCION sea diferente a Pagado, pago manual, pago manual con salida de parqueadero, entonces, se arma la trama tomando el número celular, monto de la tabla USUARIO y TIEMPO_USADO para enviar el cobro por la red GSM como se observa en la Figura 53. Enviado el pago se escanea la red GSM en busca de la cadena de caracteres "Transacción en progreso" evidenciado en la Figura 53.

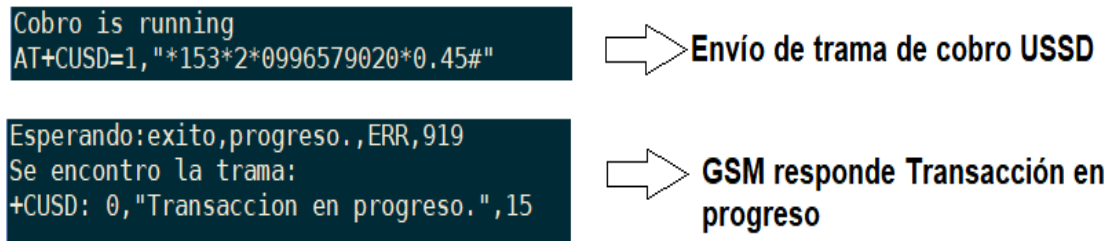


Figura 53. Proceso de envío de cobro por la red GSM.

Posteriormente se actualiza el campo ESTADO_PAGO a 2 que hace referencia a pago automático enviado por la red GSM hacia el BCE en la tabla TRANSACCION como se observa en la Figura 54.

```
MariaDB [COBRO]> SELECT * FROM TRANSACCION WHERE XID_USUARIO = 1 ORDER BY ID DESC LIMIT 1;
+-----+-----+-----+-----+-----+
| ID | XID_USUARIO | XID_TIEMPOUSADO | XID_ESTADOPAGO | DATE_TIME |
+-----+-----+-----+-----+-----+
| 35 | 1 | 35 | 2 | 2017-12-14 05:21:51 |
+-----+-----+-----+-----+-----+
```

Figura 54. Actualización de estado de pago enviado.

Posterior al envío del cobro al BCE, este envía una petición de pago USSD por la red GSM al usuario que debe realizar el pago como se muestra en la Figura 55.



Figura 55. Mensaje USSD de solicitud de pago.

Este pago puede ser exitoso o erróneo. Se válida la respuesta de parte del BCE escaneando la red GSM. En caso de obtener la respuesta de parte del BCE de “Se realizó el cobro de forma exitosa” observado en la Figura 56, se actualiza el estado de pago en la tabla TRANSACCION a 4 que hace referencia a pago automático realizado como se evidencia en la Figura 56.

```

Cobro is running
Se encontro la trama
+CUSD: 0,"Se realizo el cobro de $ 0.45 de forma exitosa. Su saldo es 43.57",15

MariaDB [COBR0]> SELECT * FROM TRANSACCION WHERE XID_USUARIO = 1 ORDER BY ID DESC LIMIT 1;
+-----+-----+-----+-----+-----+
| ID | XID_USUARIO | XID_TIEMPOUSADO | XID_ESTADOPAGO | DATE_TIME |
+-----+-----+-----+-----+-----+
| 35 |          1 |          35 |          4 | 2017-12-14 05:38:34 |
+-----+-----+-----+-----+-----+
  
```

Figura 56. Evidencia de cobro exitoso.

Además el BCE notifica al usuario que su pago fue exitoso mediante mensaje USSD y mensaje de texto a través de la red GSM, como se evidencia en la Figura 57.

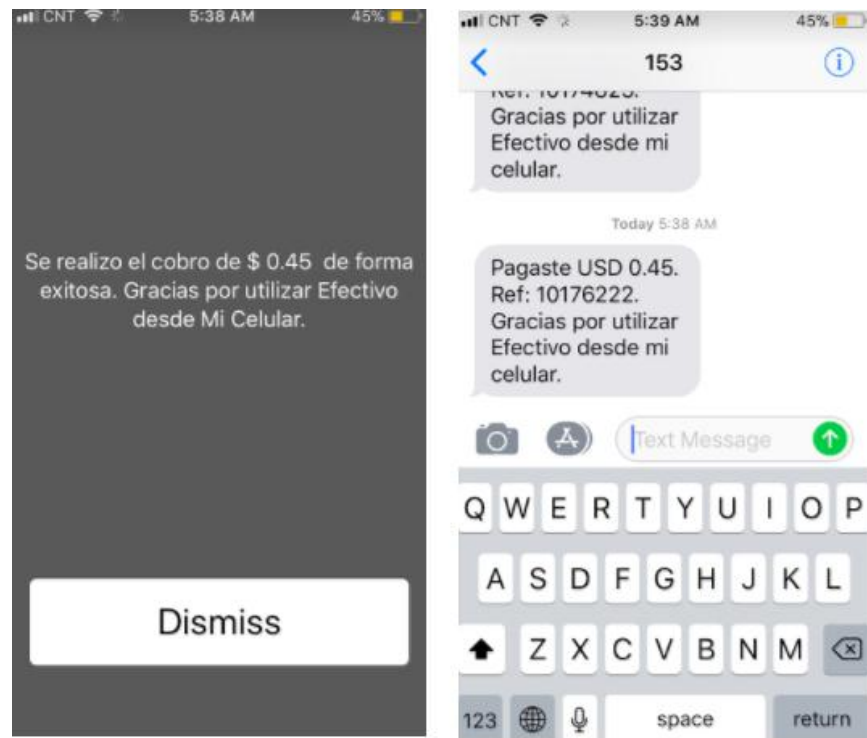


Figura 57. Notificaciones de pago exitoso al usuario por parte del BCE.

En caso de obtener la respuesta “Code 919” o “No se puede realizar la transaccion” se actualiza el estado de pago a 3 que hace referencia a error en el pago automático y se reintenta el cobro hasta un máximo de 3 veces como se evidencia en la Figura 58 a continuación.

```

MariaDB [COBRO]> SELECT * FROM TRANSACCION WHERE XID_USUARIO = 1 ORDER BY ID DESC LIMIT 1;
+-----+-----+-----+-----+-----+
| ID | XID_USUARIO | XID_TIEMPOUSADO | XID_ESTADOPAGO | DATE_TIME |
+-----+-----+-----+-----+-----+
| 35 | 1 | 35 | 3 | 2017-12-14 05:38:20 |
+-----+-----+-----+-----+-----+
pago fallido
fallida transf:UPDATE TRANSACCION SET XID_ESTADOPAGO = 3, DATE_TIME = NOW() WHERE XID_ESTADOPAGO IN (1,2,3) AND XID_USUARIO =1
pago fallido se deberia reintentar
Cobro is running
Se encontro la trama
+CUSD: 0,"No se puede realizar la transaccion.
Intentelo mas tarde.
Gracias por utilizar Efectivo desde mi celular.
Code: 919.",15

```

Figura 58. Evidencia de cobro erróneo y reintento de cobro.

El usuario recibe el mensaje de error en la transacción en su teléfono móvil como se observa en la Figura 59 a continuación.

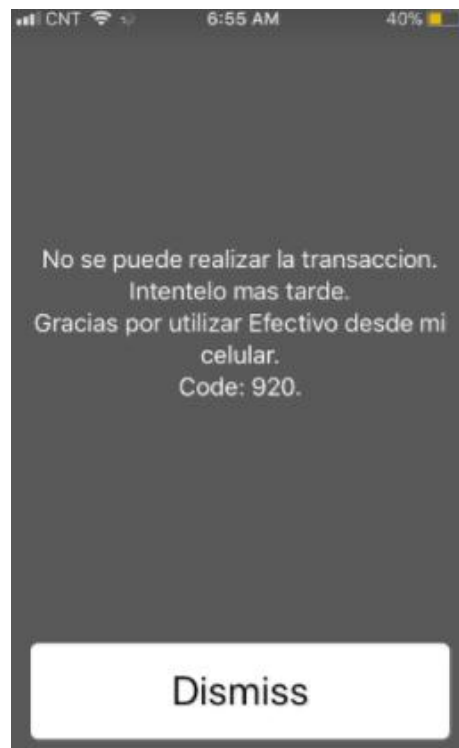


Figura 59. Notificación de transacción fallida al usuario de parte del BCE.

3.2.3.3 Fase De Pago Emergente

Esta fase es utilizada cuando existe algún inconveniente con la plataforma del BCE o a su vez cuando el usuario ha recibido una etiqueta NFC reconfigurada la cual se entrega a los usuarios que no cuenten con identificación de la universidad, esta etiqueta debe ser devuelta en la salida del usuario.

El ingreso de manera habitual presentando la etiqueta NFC otorgada por parte de la administración del parqueadero.

Al momento de realizar la salida el usuario debe acercarse a un administrador de parqueadero a cancelar manualmente el monto.

La persona encargada de administrar el parqueadero debe realizar el *login* en la página WEB <http://192.168.100.52/udlapark/>, para estas pruebas el usuario que se utilizó es “webadm” y contraseña “pass123” como se observa en la Figura 60 a continuación.



Figura 60. Ingreso a la WEB administración.

Posteriormente es redirigido a una página donde se solicita ingresar número de matrícula, la cual se encuentra en cada uno de las etiquetas de identificación de otorgadas por la universidad o para en etiquetas entregadas por la administración de parqueaderos como se observa en la Figura 61.



Figura 61. Etiquetas NFC.

Tras el ingreso del número de matrícula o identificación de etiqueta es redirigido a otra página web donde se muestra el valor a pagar con la opción de pago manual como se muestra en la Figura 62.



Figura 62. Web de ingreso de identificación y pago emergente.

Tras cancelar el valor a pagar, la persona encargada de administrar el parqueadero dará *click* en el botón pagar, posteriormente se muestra el resultado del pago en la WEB como se muestra en la Figura 63 y consecuentemente se actualiza las tablas TIEMPO_USADO y TRANSACCION en la base de datos. Se diferencia la actualización de la tabla TRANSACCION en el pago automático del pago emergente dado que el estado del pago se coloca en 5 que hace referencia a pago manual realizado como se observa en Figura 64.

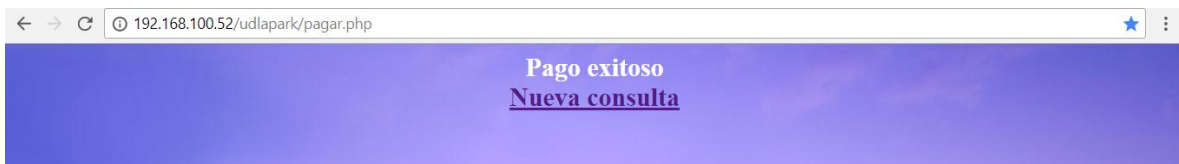


Figura 63. Mensaje resultado del pago emergente.

```
MariaDB [COBR0]> SELECT * FROM TRANSACCION WHERE XID_USUARIO = 2 ORDER BY ID DESC LIMIT 1;
+-----+-----+-----+-----+-----+
| ID | XID_USUARIO | XID_TIEMPOUSADO | XID_ESTADOPAGO | DATE_TIME |
+-----+-----+-----+-----+-----+
| 36 |          2 |          36 |          5 | 2017-12-14 07:42:08 |
+-----+-----+-----+-----+-----+
```

Figura 64. Actualización de estado de pago.

Terminado este proceso el usuario puede presentar su etiqueta NFC en el lector de salida, donde se valida que el estado del pago sea 5 ó sea pago emergente efectuado. Posteriormente se envía voltaje al circuito acoplado con el mando RF para permitir la salida del vehículo y consecuentemente se actualizará el estado de pago en la tabla “*TRANSACCION*” a 6 que hace referencia de pago manual y salida de vehículo realizado como se muestra en la Figura 65.

```
MariaDB [COBRO]> SELECT * FROM TRANSACCION WHERE XID_USUARIO = 2 ORDER BY ID DESC LIMIT 1;
+-----+-----+-----+-----+-----+
| ID | XID_USUARIO | XID_TIEMPOUSADO | XID_ESTADOPAGO | DATE_TIME |
+-----+-----+-----+-----+-----+
| 36 | 2 | 36 | 6 | 2017-12-14 07:48:36 |
+-----+-----+-----+-----+-----+
```

Figura 65. Actualización de estado de pago.

3.2.4 Resumen De Resultados Y Eventos

En la Tabla 6 se presenta los resultados obtenidos en las pruebas realizadas.

Tabla 6.

Resumen de resultados.

| Fase | Evento | Resultado exitoso | Resultado con error |
|----------|----------------------------|-------------------|---------------------|
| Ingreso | Lectura de etiqueta | 10 | 0 |
| | Escritura en base de datos | 10 | 0 |
| | Apertura de brazo mecánico | 10 | 0 |
| Consulta | Consulta de datos | 10 | 0 |
| Salida | Envío de cobro | 7 | 3 |

| | | | |
|--|--------------------|---|---|
| | Resultado de cobro | 5 | 5 |
| | Pago emergente | 5 | 0 |

En la Tabla 7 se identifican los tiempos empleados para cada prueba realizada.

Tabla 7.

Tiempos por prueba.

| | Evento | Tiempo sin error(segundos) | Tiempo con error(segundos) |
|---------|--------------------------------|----------------------------|----------------------------|
| Ingreso | Lectura de etiqueta | 0.5 | 0.5 |
| | Escritura en base de datos | 0.5 | 0.5 |
| | Apertura de brazo mecánico | 5 | 5 |
| Salida | Envío de cobro 1 reintento | 5 | 50 |
| | Resultado de cobro 1 reintento | 20 | 50 |
| | Pago emergente | 0 | 50 |
| | TOTAL | 30 | 155 |

3.2.5 Análisis De Resultados

A continuación se presenta los resultados de las pruebas realizadas en los escenarios de éxito y error.

Resultados

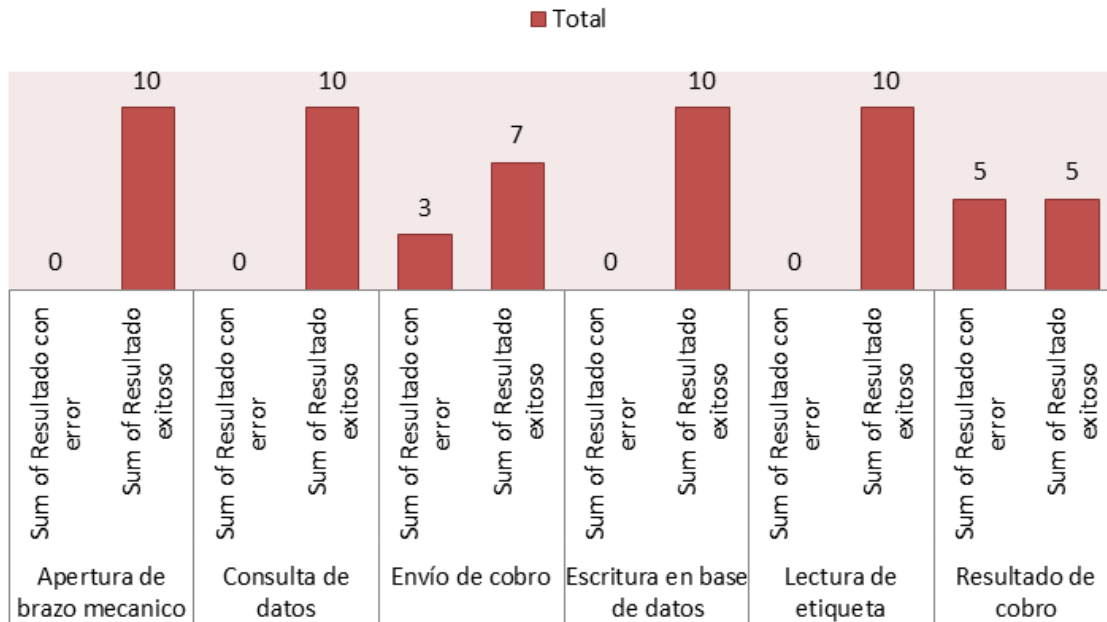


Figura 66. Resultados.

En la Figura 66 se puede observar que de las 10 pruebas realizadas el único modulo que falla es el del cobro con dinero electrónico el cual se atribuye a la disponibilidad del BCE y el factor de cobertura GSM. Estas pruebas identificaron diferentes tipos de inconvenientes al momento de realizar el cobro los cuales se identifican en la Figura 67.

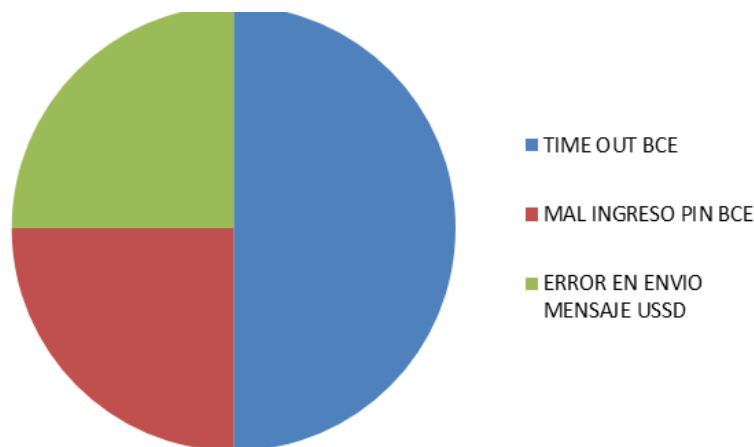


Figura 67. Errores en cobro.

Adicional, como se muestra en la Figura 68 se puede apreciar que en el mejor de los casos el tiempo empleado en ingresar y salir del parqueadero es de 30 segundos por reintento, y en escenarios con error sería de alrededor de un minuto por reintento.

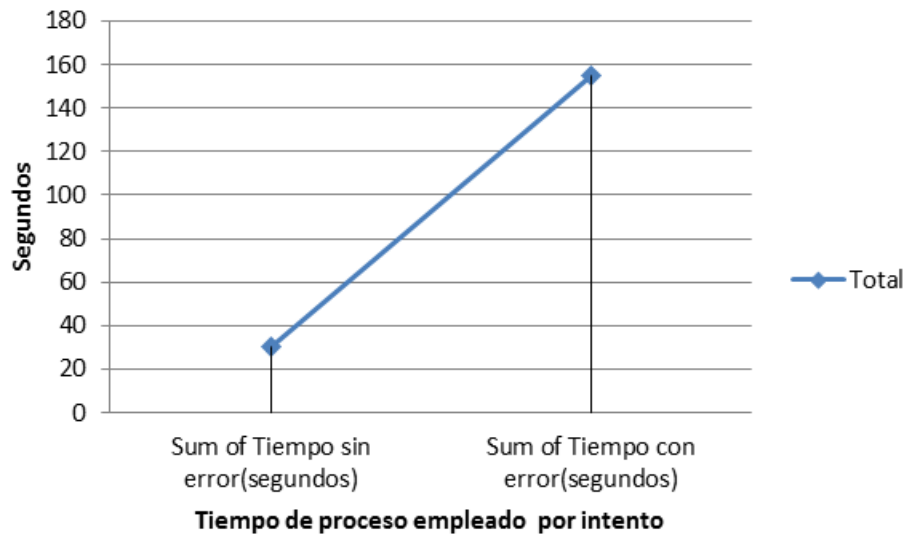


Figura 68. Tiempo utilizado por intento.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Terminado el prototipo y gracias a los conocimientos adquiridos en las fases de diseño, planificación e implementación del prototipo se tienen las siguientes conclusiones:

El prototipo de un sistema de cobro con dinero electrónico para el parqueadero de la Universidad de las Américas permite integrar conceptos de electrónica, sistemas, redes y telecomunicaciones.

El procesamiento de datos fue posible gracias a la integración del sistema operativo de software libre Raspbian. Sobre el cual se ejecuta la base de datos MariaDB, el servidor web Apache2 y código desarrollado en Python, PHP y Shell.

El prototipo integra tres tipos de comunicación inalámbrica. Red GSM para la comunicación con el BCE, NFC para la lectura de etiquetas y Wlan para acceder al servicio de visualización de data WEB.

El sistema de cobro a través de dinero electrónico desarrollado, permite realizar el cobro de los parqueadero de los estudiantes de la UDLA, a través de la solución de dinero electrónico que brinda el Banco Central del Ecuador, con lo que se consigue una importante reducción de tiempo empleado en pagos en ventanillas;

La solución desarrollada brinda un método de administración de dinero ecológico, dado que no interviene dinero físico, ni impresiones de comprobantes o facturas.

La solución desarrollada ofrece un servicio web adicional, que permite al usuario la consulta del valor por el tiempo transcurrido en el parqueadero y la forma de pago emergente en caso de existir cualquier dificultad con la plataforma del BCE.

La fase de pruebas evidenció errores que afectan directamente al cobro con dinero electrónico a través de la red GSM. Los errores comunes son:

Ingreso erróneo de clave de BCE, indisponibilidad de la plataforma del BCE y caducidad del tiempo que tiene el usuario para ingresar la clave de para efectuar el pago.

El modo de pago empleado para este desarrollo es el servicio de dinero electrónico del BCE. Y gracias a que el prototipo cuenta con una base de datos relacional es posible ampliar el cobro a través de otras formas de dinero electrónico como debito directo de cuentas bancarias, pago de parqueadero prepago entre cualquier otro modo de cobro electrónico existente.

El cobro se puede realizar desde cualquier terminal móvil que cuente con tecnología GSM sin la necesidad de conectividad a internet.

4.2 Recomendaciones

Las recomendaciones mencionadas a continuación deben ser tomadas en cuenta para la correcta operación del prototipo y para el mejoramiento del mismo en futuros proyectos de titulación, a partir de la base del presente proyecto.

Se recomienda tener en consideración que el mando RF de apertura de brazos mecánicos utiliza su propia alimentación mediante batería y estas pueden agotarse.

Es posible mejorar el prototipo incorporando un sistema de lectura de placas de autos, incorporado al Raspberry Pi, de esta manera se mejoraría el tiempo al ingreso del parqueadero sin la necesidad de lecturas de etiquetas NFC.

El presente prototipo realiza el cobro mediante la tecnología inalámbrica USSD sobre la red GSM, como toda tecnología inalámbrica es propensa a fallos, por lo que es posible mejorar la fiabilidad consumiendo un servicio web directamente del BCE.

Es posible mejorar el subsistema de visualización de data en el módulo de administrador, para que permita visualizar reportaría de las transacciones

realizadas. Además de que permita visualizar el saldo que se tiene en la cuenta de dinero electrónico integrada al prototipo.

El modulo WEB de administración no cuenta con cifrado SSL por lo que es aconsejable la adquisición de un certificado y la utilización del protocolo HTTPS.

Se recomienda implementar la solución incorporando otras opciones de pago electrónico.

Se recomienda realizar mantenimientos a los brazos mecánicos y demás estructura de control de acceso a los parqueaderos para que permitan un sistema integral y funcional.

Referencias

- Balboa. (2002). Servicios portadores. MADRID: PEARSON EDUCACIÓN, S.A.
- Banco Central del Ecuador. (2014). Regulación N°. 055. Recuperado el 07 de noviembre de 2017, de https://www.bce.fin.ec/images/transparencia/informacion_legal/documentos/regulaciones2014/regulacion_0552014.pdf
- Communitic International. (2013). La tecnología NFC. Recuperado de <http://es.ccm.net/faq/10985-la-tecnologia-nfc#tres-modos-de-funcionamiento>
- Dakel. (2016). Informatica centralizada. Recuperado de <http://www.dakel.com/2016/12/27/informatica-centralizada-todo-son-ventajas-para-la-empresa>
- Dordoigne, J. (2015). Redes informáticas nociones fundamentales. Barcelona: Ediciones ENI.
- ETSI. (1997). Estandar GSM03.90. Recuperado el 29 de octubre de 2017, de http://www.etsi.org/deliver/etsi_gts/03/0390/05.00.00_60/gsm03_90v050000p.pdf
- ETSI. (1997). Estandar GSM02.90. Recuperado el 29 de octubre de 2017, de http://www.etsi.org/deliver/etsi_gts/02/0290/05.01.00_60/gsm02_90v050100p.pdf
- Hernandez, A. (2016). Auc Proceso de autenticación. Recuperado de <http://slideplayer.es/slide/4638783>
- Huawei. (2016). Manual de usuario Huawei E3531. Recuperado de <https://usermanual.wiki/Huawei/DDataWebpageDownloadE3531QSE3531DEpdf.1301444217>
- IBM. (2017). Bases de datos SQL. Recuperado el 11 de noviembre de 2017, de <http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/reldb>
- IBM. (2017). Red heterogénea. Recuperado el 03 de octubre de 2017, de https://www.ibm.com/support/knowledgecenter/es/ssw_aix_61/com.ibm.aix.install/nim_networks_hetero.htm

- informática. (2013). historia de la informática. Recuperado el 8 de 11 de 2017, de <https://histinf.blogs.upv.es/>
<https://histinf.blogs.upv.es/2013/12/18/raspberry-pi>
- ISO/IEC. (2002). Estandar ISO/IEC 14448. Recuperado el 08 de noviembre de 2017, de <http://www.atmel.com/images/doc2056.pdf>
- ISO/IEC. (2013). Estandar ISO 18092. Recuperado el 08 de noviembre de 2017, de http://standards.iso.org/ittf/PubliclyAvailableStandards/c056692_ISO_IEC_18092_2013
- Ecuador Legal.(2015). Evolución de los medios de pago en el Ecuador. Recuperado el 07 de Noviembre de 2017, de Ecuador legal online: <http://www.ecuadorlegalonline.com/dinero-electronico/evolucion-medios-de-pago-ecuador>
- Mariadb. (2017). Base de datos mariadb. Recuperado el 8 de noviembre de 2017, de mariadb: <https://mariadb.com/kb/es/about-mariadb>
- Mrdesc. (2013). Componentes GSM. Recuperado de sdrlatino: <https://sdrlatino.wordpress.com/2013/06/28/introduccion-a-la-tecnologia-de-acceso-gsm>
- NFC Forum. (2010). NFC Technical Specifications. Recuperado el 20 de noviembre de 2017, de http://members.nfc-forum.org/specs/spec_list
- NFC. (2012). Tipos de comunicación NFC. Recuperado el 20 de noviembre de 2017, de NFC Mundo: <https://mundonfc.wordpress.com/2012/02/09/como-funciona-el-nfc>
- PHP. (2017). Lenguaje de programación PHP. Recuperado el 20 de noviembre de 2017, de <http://php.net/manual/es/intro-what-is.php>
- RASPBERRY PI FOUNDATION. (2017). Especificaciones Raspberry Pi. Recuperado el 11 de noviembre de 2017, de RASPBERRY PI: <https://www.raspberrypi.org>
- Rey. (1998). Telecomunicaciones móviles 2nda edición. Barcelona: FOINSA.
- Selena, L. (2017). Equipo móvil y tarjeta SIM. Obtenido de <https://drfone.wondershare.com/sim-unlock/how-to-unlock-iphone-5-to-any-network.html>

- Universidad Politecnica de Madrid. (2007). Comunicaciones Moviles GSM. Recuperado el 03 de octubre de 2017, de <http://ocw.upm.es/teoria-de-la-senal-y-comunicaciones-1/comunicaciones-moviles-digitales/contenidos/Presentaciones/GSM-07.pdf>
- Vertice. (2009). Diseño basico de paginas weben HTML. España: Editorial Vértice.
- Villegas. (2014). g Modelo cliente servidor. Recuperado de <http://villegitz.blogspot.com/2014/04/informatica-en-computacion-2.html>
- Wilson. (2013). Manual de módulo PN532. Recuperado el 8 de noviembre de 2017, de https://dangerousthings.com/wp-content/uploads/PN532_Manual_V3-1.pdf
- Wolf. (2015). Fundamento de sistemas operativos. Fundamento de sistemas operativos. Mexico DF: Universidad Nacional Autonoma de Mexico.

ANEXOS

ANEXO 1

Código de archivo in_parking.py:

```
import subprocess #Se importa librerias necesarias
import time
import os

from sql import *
from rf_control import *

def nfc_raw(): #Esta clase define que modulo NFC se va a utilizar
    lines=subprocess.check_output("/usr/bin/read-in",
stderr=open('/dev/i2c-0','w'))

    return lines

def read_nfc(): #Esta clase realiza lecturas a traves del modulo NFC

    lines=nfc_raw()

    return lines

#Inicia bucle de lectura NFC

try:

    while True:

        myLines=read_nfc()

        buffer=[]

        for line in myLines.splitlines():

            line_content=line.split()

            if(not line_content[0] =='UID'):

                pass

            else:

                buffer.append(line_content)

        str=buffer[0]

        id_str=str[2]+str[3]+str[4]+str[5]
```

```

    print (id_str)

    tagid = "%s" % tag_id(id_str)

    if(tagid != 'None'):

        utid = "%s" % usedt_id(id_str)

        payed = "%s" % manual_pay(tagid)

        if(utid == 'None' and payed == 'None'):

            in_parking(id_str) #realiza llamada de clase
in_parking con el id del tag como parametro

            time.sleep(2)

            print "ejecutado abriendo puerta"

            open_door()

            time.sleep(10)

            print "cerrando puerta"

            close_door()

        else:

            print "Ya tiene un ingreso activo"

    else:

        print "usuario no existe"

except KeyboardInterrupt:

    pass

```

ANEXO 2

Código de archivo exit_parking.py

```

import subprocess #se importa librerias necesarias

import time

import os

from gsm_transactions import gsm_inicio

from sql import *

def nfc_raw(): #Esta clase define que modulo se va a utilizar

```

```
        lines=subprocess.check_output("/usr/bin/read-exit",
stderr=open('/dev/i2c-1','w'))
```

```
    return lines
```

```
def read_nfc(): #Esta clase realiza escaneo mediante el modulos NFC
```

```
    lines=nfc_raw()
```

```
    return lines
```

```
#Inicia bucle de cobro
```

```
try:
```

```
    gsm_status=gsm_inicio()
```

```
    print gsm_status
```

```
    if (gsm_status == '1'):
```

```
        while True:
```

```
            myLines=read_nfc()
```

```
            buffer=[]
```

```
            for line in myLines.splitlines():
```

```
                line_content=line.split()
```

```
                if(not line_content[0] == 'UID'):
```

```
                    pass
```

```
                else:
```

```
                    buffer.append(line_content)
```

```
                    str=buffer[0]
```

```
                    id_str=str[2]+str[3]+str[4]+str[5]
```

```
                    print (id_str)
```

```
                    tagid = "%s" % tag_id(id_str)
```

```
                    print tagid
```

```
                    if (tagid != 'None'):
```

```
                        utid = "%s" % usedt_id(id_str)
```

```
                        print utid
```

```

        if (utid != 'None'):
            print "esto no debe estar vacio"+id_str
            salida_parqueadero(id_str)
        else:
            print "no existe ingreso"
    else:
        print "no existe usuario"

except KeyboardInterrupt:
    pass

```

ANEXO 3

Código archivo gsm_transactions.py

```

import RPi.GPIO as GPIO #se incluye librerias necesarias
import serial
import time, sys
import datetime

def wait_response(resp,resp2,resp3,err):
    reply = ser.read(ser.inWaiting())
    reply = str(reply)
    resp = str(resp)
    resp2 = str(resp2)
    resp3 = str(resp3)
    err = str(err)
    tout = 0
    print "Esperando:"+ resp +", "+resp2+", "+resp3+", "+err

while (reply != resp):

```

```

if (tout <= 800000):
    tout = tout +1
    reply = ser.read(ser.inWaiting())
    if re.search(resp3 , reply):
        print "Se encontro en la trama: "+reply
        resp = reply
        return "0"
    if re.search(resp2 , reply):
        print "Se encontro la trama:"+reply
    if re.search(resp , reply):
        print "Se encontro la trama"+reply
        resp = reply
        return "1"
    if re.search(err , reply):
        print "Se encontro la trama"+reply
        resp = reply
        return "0"
else:
    ser.write("AT+CUSD=2")
    resp=reply
    print "Time out bye!"
    return "0"

```

```
SERIAL_PORT="/dev/ttyUSB0"
```

```
ser=serial.Serial(SERIAL_PORT, baudrate = 9600, timeout = 5)
```

```
#def gsm_saldo(): #clase de consulta de saldo
```

```
#    ser.write("AT^USSDMODE=0\r")
```

```
#    wait_response("OK","null","null","ERR")
```



```

#     time.sleep(1)
#     ser.write("AT+CUSD=1,\"*153*1*1*0966#\r")
#     time.sleep(1)
#     wait_response("ponible","OK","null","ERR")

def gsm_cobro(num, monto):
    num = str(num)
    monto = str(monto)
    ser.write("AT^USSDMODE=0\r")
    time.sleep(2)
    trama = "AT+CUSD=1,\"*153*2*"+num+"*"+monto+"#\r"
    ser.write(trama)
    print trama
    status=wait_response("exito","progreso.", "ERR", "919")
    print "Retorno estado cobro : "+str(status)
    return status

def gsm_inicio():
    ser.write("AT+CREG?\r")
    status=wait_response("OK", "null", "ERR", "null")
    print "Retorno estado GSM: "+str(status)
    return status

```

ANEXO 4

Código de archivo rf_control.py

```

import RPi.GPIO as GPIO

import time

def open_door():
    GPIO.setmode(GPIO.BCM)

    GPIO.setup(11, GPIO.OUT) ## GPIO 11 como salida

    GPIO.output(11, True)

```

```
time.sleep(2)

GPIO.output(11, False)

def close_door():

    GPIO.setmode(GPIO.BCM)

    GPIO.setup(26, GPIO.OUT)

    GPIO.output(26, True)

    time.sleep(2)

    GPIO.output(26, False)
```

ANEXO 5

Código de archivo sql.py

```
import MySQLdb #se incluye librerias necesarias

import os

import time

from rf_control import *

from gsm_transactions import *
```

```

def tag_id(tag):#clase consulta id de usuario de id de etiqueta NFC

    db = MySQLdb.connect("localhost","pay_check","pass123","COBRO")

    cursor = db.cursor()

    a = "select ID from USUARIO where TAG = '"

    sql = a + str(tag) + "'"

    print sql

    cursor.execute(sql)

    data = cursor.fetchone()

    cursor.close()

    db.close()

    return data

def usedt_id(tag): #Clase consulta si existe ya un ingreso

    db = MySQLdb.connect("localhost","pay_check","pass123","COBRO")

    cursor = db.cursor()

    id_u = "%s" % tag_id(tag)

    sql = "select MAX(XID_TIEMPOUSADO) from TRANSACCION WHERE
XID_ESTADOPAGO <> 4 AND XID_ESTADOPAGO <> 6 AND XID_USUARIO = "+ id_u

    print sql

    cursor.execute(sql)

    data = cursor.fetchone()

    cursor.close()

    db.close()

    return data

def manual_pay(id_u): #Clase valida pago manual

    db = MySQLdb.connect("localhost","pay_check","pass123","COBRO")

    cursor = db.cursor()

    sql = "SELECT MAX(XID_TIEMPOUSADO) FROM TRANSACCION WHERE
XID_ESTADOPAGO = 5 AND XID_USUARIO =" + id_u

```

```
cursor.execute(sql)

data = cursor.fetchone()

cursor.close()

db.close()

return data
```

```
def amount(tag): #Clase calcula balor final a pagar

db = MySQLdb.connect("localhost","pay_check","pass123","COBRO")

cursor = db.cursor()

id_u = "%s" % tag_id(tag)

tran_id = "%s" % usedt_id(tag)

famount = "SELECT ROUND(ROUND(TIMESTAMPDIFF(MINUTE, TIEMPO_ENTRADA,
TIEMPO_SALIDA)/ 60, 2)*0.50, 2) FROM TIEMPO_USADO WHERE XID_USUARIO = "+
id_u + " AND ID = "+tran_id

cursor.execute(famount)

data = cursor.fetchone()

cursor.close()

db.close()

return data
```

```
def float_amount(tag): #clase consulta valor variable a pagar

db = MySQLdb.connect("localhost","pay_check","pass123","COBRO")

cursor = db.cursor()

id_u = "%s" % tag_id(tag)

famount = "SELECT ROUND(ROUND(TIMESTAMPDIFF(MINUTE, TIEMPO_ENTRADA,
NOW())/ 60, 2)*0.50, 2) FROM TIEMPO_USADO WHERE XID_USUARIO = "+ id_u + "
AND MONTO IS NULL"

cursor.execute(famount)

data = cursor.fetchone()

cursor.close()

db.close()
```

```

return data

def in_parking(tag): #realiza proceso de ingreso al parqueadero

    db = MySQLdb.connect("localhost","pay_check","pass123","COBRO")

    cursor = db.cursor()

    id_u = "%s" % tag_id(tag)

    used_time = "INSERT INTO TIEMPO_USADO (XID_USUARIO, TIEMPO_ENTRADA)
VALUES (" + id_u + ", NOW())"

    cursor.execute(used_time)

    db.commit

    sql = "SELECT MAX(ID) FROM TIEMPO_USADO WHERE XID_USUARIO = " + id_u
+" AND TIEMPO_SALIDA IS NULL"

    cursor.execute(sql)

    id_ut = "%s" % cursor.fetchone()

    print id_ut

    transaction = "INSERT INTO TRANSACCION (XID_USUARIO, XID_TIEMPOUSADO,
XID_ESTADOPAGO, DATE_TIME) VALUES (" + id_u + "," + id_ut + ",1,NOW())"

    cursor.execute(transaction)

    db.commit()

    cursor.close()

    db.close()

def validar_pago(status, id_u): #Clase valida pago

    db = MySQLdb.connect("localhost","pay_check","pass123","COBRO")

    cursor = db.cursor()

    id_u = str(id_u)

    print "dentro de validar pago" +id_u

    status = int(status)

    print "dentro de validar pago: "

    print status

    if status == 1:

```

```

        sql = "UPDATE TRANSACCION SET XID_ESTADOPAGO = 4, DATE_TIME =
NOW() WHERE XID_ESTADOPAGO IN (1,2,3) AND XID_USUARIO =" + id_u

        cursor.execute(sql)

        db.commit()

        cursor.close()

        db.close()

        a=1

        return a

    if status == 0:

        print "pago fallido"

        sql2 = "UPDATE TRANSACCION SET XID_ESTADOPAGO = 3, DATE_TIME =
NOW() WHERE XID_ESTADOPAGO IN (1,2,3) AND XID_USUARIO =" + id_u

        print "fallida tranf:" + sql2

        cursor.execute(sql2)

        db.commit()

        cursor.close()

        db.close()

        a=0

        return a

def pago(id_u,amount):#clase realiza pago

    db = MySQLdb.connect("localhost","pay_check","pass123","COBRO")

    cursor = db.cursor()

    id_u = str(id_u)

    monto = str(amount)

    num = "SELECT TELEFONO FROM USUARIO WHERE ID = " + id_u

    cursor.execute(num)

    num = cursor.fetchone()

    num = "%s" % num

    print num

```

```

estado = "%s" % manual_pay(id_u)

print "imprimiendo estado de transaccion: " + estado

estado = str(estado)

if (estado == 'None'):

    status=gsm_cobro(num,monto)

    tran = "UPDATE TRANSACCION SET XID_ESTADOPAGO = 2, DATE_TIME =
NOW() WHERE XID_ESTADOPAGO IN (1,2,3) AND XID_USUARIO = "+ id_u

    cursor.execute(tran)

    db.commit()

    cursor.close()

    db.close()

    print "imprimiendo estado 1 cobro exitoso 0 error: "

    print status

    return status

else:

    print "Pago manual realizado"

    tran = "UPDATE TRANSACCION SET XID_ESTADOPAGO = 6, DATE_TIME =
NOW() WHERE XID_ESTADOPAGO = 5 AND XID_USUARIO = "+id_u

    cursor.execute(tran)

    db.commit()

    cursor.close()

    db.close

    status = 1

    return status

def valida_parqueadero(tag):#clase realzia salida y validaciones de
parqueadero

    db = MySQLdb.connect("localhost","pay_check","pass123","COBRO")

    cursor = db.cursor()

```

```

id_u = "%s" % tag_id(tag)

id_tran = "%s" % usedt_id(tag)

print "id de transaccion para modificar usedtime" +id_tran

exit_time = "UPDATE TIEMPO_USADO SET TIEMPO_SALIDA = NOW() WHERE
XID_USUARIO =" + id_u + " AND ID = "+id_tran

cursor.execute(exit_time)

db.commit()

tamount = "%s" % amount(tag)

fill_amount="UPDATE TIEMPO_USADO SET MONTO = " + tamount + " WHERE
XID_USUARIO = " + id_u + " AND ID = "+id_tran

cursor.execute(fill_amount)

db.commit()

db.close()

pstatus=pago(id_u,tamount)

cont = 1

while (cont <= 3):

    print "imprimiendo contador"

    print cont

    if cont <= 3:

estado=validar_pago(pstatus,id_u)

estado = int(estado)

if estado == 1:

    time.sleep(2)

    print "pago exitoso abriendo salida"

    open_door()

    cont = 4

if estado == 0:

    print "pago fallido se deberia reintentar"

    pstatus=pago(id_u,tamount)

```



```
        cont = int(cont) + 1
        print "reintento #:"
    print "final de contador"
    print cont
```

ANEXO 6

Código de inicio de programa

```
#!/bin/bash
#JG
while : ;do
p1=`ps -ef |grep "exit_parking.py" |grep -v "grep" |awk '{print $2}'`
p2=`ps -ef |grep "in_parking" |grep -v "grep" |awk '{print $2}'`
p3=`ps -ef |grep "/usr/bin/read-in" |grep -v "grep" |awk '{print $2}'`
p4=`ps -ef |grep "/usr/bin/read-exit" |grep -v "grep" |awk '{print $2}'`

echo $p1,"",$p2,"",$p3,"",$p4
if [ -z "$p1" ] && [ -z "$p2" ]; then
    python bin/exit_parking.py &
    python bin/in_parking.py &
    echo "Cobro started"

else
    if [ -z "$p1" ] || [ -z "$p2" ]; then
        sh kill_demond_cobro.sh
    else
        echo "Cobro is running"
    fi
fi
sleep 3
done
```

```
#!/bin/bash

#JG

pids=`ps -ef |grep -e "exit_parking.py" -e "in_parking" -e
"/usr/bin/read-in" -e "/usr/bin/read-exit" |grep -v "grep" |awk '{print
$2}'`

if [ -z "$pids" ]; then

    echo "Cobro is not executed"

else

    kill -9 $pids

    echo "Cobro stopped"

fi
```

