

Reingeniería en firmware de controladora AccesScan v 4.0

JAIME ENRIQUE PEDRAZA SUAREZ
EDWIN ALEJANDRO RODRÍGUEZ BELTRÁN

Director
Dianalin Neme Prada

UNIVERSIDAD ANTONIO NARIÑO
INGENIERÍA EN SISTEMAS
ESPECIALIZACION EN INGENIERIA DE SOFTWARE
BOGOTA D.C
2020

Contenido

Introducción	9
1. Formulación y descripción del problema.....	11
1.1 Presentación del problema	11
1.2 Formulación del problema	12
1.3 Síntesis	15
2. Objetivo General.....	16
3. Objetivos Específicos.....	16
4. Marco de Referencia	17
4.1 Estado del arte.....	17
4.2 Impacto	21
4.3 Componente de innovación.....	21
4.4 Marco teórico	22
5. Metodología	30
6. Proceso de Software.....	31
6.1 Requerimientos funcionales.....	31
6.2 Requerimientos no funcionales.....	33
6.3 Diseño y arquitectura	34
6.3.1 Diagrama de despliegue.....	34
6.3.2 Caso de uso arquitectura relevante	35
6.3.3 Diagrama de secuencia	36
6.3.4 Arquitectura de alto nivel.....	37
6.4 Construcción	38

6.5 Pruebas	48
6.6 Instalación y configuración.....	52
7. Anexos	55
A. Aplicación de metodología	55
B. Herramientas de seguimiento	63
C. Protocolo de encriptación de claves	64
D. Resultados obtenidos.....	65
• Disminución de los servicios técnicos.	65
• Rapidez en adaptarnos a las necesidades del cliente.	68
E. Análisis de gestión de negocio	69
8. Conclusiones	71
9. Referencias – Bibliografía	73

Tabla de figuras

<i>Figura 1.</i> Controladora AccesScan versión 4. Autoría propia.....	26
<i>Figura 2.</i> Transporte de tarjetas inteligentes Lintech F3. Tomada de (Automatic Card Dispenser Parking Lot System Dispenser, s. f.).....	27
<i>Figura 3.</i> Pantallas informativas del expedidor de tarjeta, se presentan tres de los modelos adaptables. Autoría propia.	27
<i>Figura 4.</i> Lector de tarjetas Mifare ZKTeco. Adaptada de (ZKTECO PROID10BM, s. f.)	28

<i>Figura 5.</i> Diagrama de despliegue, conexión de la controladora con los diferentes equipos y sistemas integrados en los expedidores de tarjetas inteligentes. Autoría propia	35
<i>Figura 6.</i> Diagrama de caso de uso de la operación normal del sistema. Autoría propia.	36
<i>Figura 7.</i> Diagrama de secuencia para el manejo de tarjetas de visitante o rotación en el sistema. Autoría propia.	37
<i>Figura 8.</i> Diagrama de secuencia manejo de tarjetas de proximidad con ingreso por transporte de tarjetas. Autoría propia.....	37
<i>Figura 9.</i> Proceso de registro de ingresos al sistema de control vehicular mediante la expedición de tarjetas operado por la controladora AccesScan. Autoría propia.	38
<i>Figura 10.</i> Código Secuencial. Secuencia de lectura de uno de los bloques de la tarjeta Mifare. Autoría propia.	40
<i>Figura 11.</i> Declaración de variables y bits. Muestra de la gran cantidad de variables declaradas en el código anterior. Autoría propia.	40
<i>Figura 12.</i> Funciones duplicadas. Detalle del uso de la misma función para realizar la misma acción de grabación en las tarjetas Mifare. Autoría propia.	41
<i>Figura 13.</i> Control actual de versiones. Cada carpeta corresponde a un solo cliente, en este se pueden encontrar más de un firmware almacenado. Autoría propia.	42
<i>Figura 14.</i> Árbol del proyectó, estructura firmware anterior. Autoría propia.	43
<i>Figura 15.</i> Implementación de máquinas de estado finito en el nuevo firmware de la controladora. Autoría propia.	44
<i>Figura 16.</i> Definición de los estados y tareas en el nuevo firmware de la controladora. Autoría propia.	45

<i>Figura 18.</i> Comparativa de la declaración de variables en los firmwares, antes y después del proceso de reingeniería. Autoría propia.....	46
<i>Figura 17.</i> Representación de las máquinas de estado finito. Autoría propia.	46
<i>Figura 19.</i> Dispositivos presentes en la controladora. Autoría propia.	47
<i>Figura 20.</i> Módulos implementados en el proyecto. Autoría propia	48
<i>Figura 21.</i> Segmento de debug realizado en los procesos de pruebas y seguimiento. Autoría propia.	50
<i>Figura 22.</i> Repositorio de versiones en Git y GitHub. Autoría propia.....	51
<i>Figura 23.</i> Aplicaciones para la generación de las claves de registro. Autoría propia.....	53
<i>Figura 24.</i> Acceso al módulo de programación mediante la generación de llaves de acceso. Autoría propia.	54
<i>Figura 25.</i> Primer caso de uso, requerimiento de funcionalidad básica de la controladora. Autoría propia.	57
<i>Figura 26.</i> Segundo caso de uso, mensuales por boca. Autoría propia.	60
<i>Figura 27.</i> Tercer caso de uso, validación de horarios de mensuales. Autoría propia.....	62
<i>Figura 28.</i> Captura de proceso implementado en software Project. Diseño propio.	63
<i>Figura 29.</i> Tablero de software Trello implementado durante el proyecto, Adaptado de (Hardware Trello, s. f.).....	63
<i>Figura 30.</i> Lista de chequeo implementado en Trello para el seguimiento de las operaciones a realizar por el equipo. Adaptado de (Hardware Trello, s. f.).	64

<i>Figura 31.</i> Calculadora de eventos de Shannon, probabilidades de acceso al módulo de configuración mediante el ingreso de datos al sistema. Tomado de (Calculadora en línea: Entropía de Shannon, s. f.).....	65
<i>Figura 32.</i> Calculadora de Shannon. Tomado de (Calculadora en línea: Entropía de Shannon, s. f.)	65
<i>Figura 33.</i> Detalle de tarjetas reportadas como perdidas. Autoría propia.	67
<i>Figura 34.</i> Detalle de tarjetas reportadas como perdidas. Autoría propia.	67
<i>Figura 35.</i> Detalle de tarjetas reportadas como perdidas. Autoría propia.	67
<i>Figura 36.</i> Detalle de tarjetas reportadas como perdidas. Autoría propia.	67
<i>Figura 37.</i> Reporte de tarjetas atascadas en los transportes, la entrada 2 corresponde a un ingreso actualizado, mientras que la entrada 3 hace parte de un equipo antiguo. Autoría propia.	68
<i>Figura 38.</i> Reporte de tarjetas atascadas en los transportes, la entrada 2 corresponde a un ingreso actualizado, mientras que la entrada 3 hace parte de un equipo antiguo. Autoría propia.	68
<i>Figura 39.</i> Reporte de tarjetas atascadas en los transportes, la entrada 2 corresponde a un ingreso actualizado, mientras que la entrada 3 hace parte de un equipo antiguo. Autoría propia.	68
<i>Figura 40.</i> Reporte de tarjetas atascadas en los transportes, la entrada 2 corresponde a un ingreso actualizado, mientras que la entrada 3 hace parte de un equipo antiguo. Autoría propia.	68

Figura 41. Grafica demostrativa de la reducción de costos obtenidos a partir del proceso de actualización realizado, información suministrada por el departamento contable de la compañía. Autoría propia.....71

Introducción

La implementación de nuevas tecnologías en los sistemas de acceso vehicular y peatonal desarrollados por la compañía Security Access FB requieren de muchas horas de dedicación y estudio, especialmente a la hora de requerir algún tipo de modificación en los programas heredados, en nuestro caso puntual a los relacionados con el manejo directo de las funciones operativas y funcionales de las controladoras diseñadas por la compañía.

El requerir de un programa para controlar dispositivos electrónicos, que inicialmente no fue estructurado adecuadamente o pensado en permitir la escalabilidad de sus servicios ni en la portabilidad de sus componentes, han generado un gran retraso en las entregas solicitadas por parte de la gerencia de la compañía, procesos que a nivel funcional no requerían más que unos pocos días para su despliegue y puesta en marcha, pasaron a ser tareas de semanas e inclusive de meses para su entrega y validación.

Por estas y muchas razones más, el equipo de desarrollo de la compañía en cabeza del ingeniero de hardware se dio a la tarea de buscar un método de mejora no solo en los procesos de desarrollo del software de los microcontroladores, también en los diferentes proyectos y servicios brindados en la actualidad por la compañía.

Es por esto que, mediante la implementación de los conocimientos adquiridos a lo largo del proceso académico y la visión de empresa adquirida por el equipo de trabajo, se realizó la reestructuración del firmware de la controladora AccesScan para el manejo del expedidor de tarjetas referencia Lintech F3. Este proceso se llevó a cabo tomando como punto de partida la operación actual del expedidor de tarjetas, el cual ha sido el mismo durante varios años y que ha sido implementando la compañía Security Access FB a nivel nacional.

La implementación de metodologías de código limpio, software elegante y de arquitectura en la programación de la controladora permitirán generar cambios a futuro de manera ágil y clara, todo mediante el desacople de las rutinas de manejo de los equipos electrónicos que pese a ser parte del conjunto funcional, no deben tener el control total de su operación final.

Otra característica obtenida del proceso implementado es contar con un sistema de operación global, al permitir la configuración de sus parámetros funcionales directamente en la instalación, disminuyendo esto la cantidad de programas generados en el departamento de desarrollo y optimizando las tareas de mantenimiento, el proceso de configuración de las controladoras será restringido a personal externo a la compañía. El acceso a las opciones de configuración de las controladoras se realizará mediante la autenticación del usuario en la App diseñada para este fin. En este programa, el usuario deberá ingresar la clave de acceso generada por la controladora a lo cual la aplicación retornara la contraseña de ingreso para ser digitalizada, esta clave y contraseña de ingreso es generada de manera aleatoria por el programa de autenticación.

Los programas se han realizado bajo estructura de programación, C para la generación del firmware de la controladora AccesScan y Android para la App de generación de llaves.

Todo el proceso de ejecución del proyecto se ha realizado siguiendo metodologías de desarrollo XP y Kanban, este proceso se ha llevado a cabo implementándolo en la actual controladora diseñada y ensamblada en las instalaciones de la compañía Security Access FB.

1. Formulación y descripción del problema

1.1 Presentación del problema

La dependencia del personal involucrado en los desarrollos de la compañía era un hecho que no desconocían las directivas de la compañía Security Access FB, pero el contar desde los inicios de operación con el mismo ingeniero de desarrollo en hardware de una u otra manera atenuaron los problemas que se presentaban al momento de requerir cualquier tipo de modificación o la adecuación de un nuevo dispositivo en la controladora de acceso AccesScan.

Pero como todo en la vida en algún momento se presentarán cambios, lo que ocurrió en la compañía, generando una reestructuración de personal de diferentes áreas incluida el área de desarrollo, en este cambio de personal se encontraba el ingeniero de desarrollo electrónico, lo cual para las operaciones esenciales y básicas no presentarían ningún tipo de alarma, pero al momento de tener que realizar el cambio de tecnologías en los dispositivos electrónicos que se integraban con la tarjeta de control AccesScan lo cual significaba un cambio de gran nivel en gran parte del firmware fue evidente la dependencia que se tenía de esta persona, no solo por sus años de experiencia o por el conocimiento de los sistemas ya desarrollados, la dependencia se evidencio al momento de tener que reestructurar el código heredado y tener que desacoplar muchas funciones que manejaban los procesos de lectura y grabación de las tarjetas inteligentes.

El problema como tal radico al no contar con librerías independientes para el manejo de funciones o dispositivos en el sistema, al apreciar por parte del nuevo ingeniero encargado de soportar el sistema, que toda la programación desarrollada hasta ese momento en la

controladora no podría ser migrada a nuevos dispositivos y que de continuar con la implementación de cambios en el sistema este podría empezar a presentar fallas de todo tipo.

1.2 Formulación del problema

La compañía Security Access FB desde sus inicios ha brindado soluciones integrales en el mercado de los sistemas de control de acceso vehicular, los productos ofrecidos por la compañía en su gran mayoría han sido desarrollados en la compañía y cada uno de ellos se encarga de realizar diferentes funciones dentro del paquete operativo que pueda ser brindado a los diferentes clientes presentes en el mercado, es por esto que el sistema de control de acceso vehicular varía según los requerimientos de cada uno de los clientes pero en esencia se manejan dos tipos de operación, local o remota.

Donde el tipo de operación local es conformado por:

- un equipo de cómputo
- un lector de tarjetas inteligente o de código de barras
- una impresora térmica

básicamente una terminal de ventas (POS)¹.

Pero si la operación es del tipo remoto, el sistema integrara diferentes dispositivos para el control de las diferentes labores a realizar, equipos de control de acceso, control de salida, puntos de pago automático y sistemas de video para el registro de los automotores. Es en este segmento del mercado en el cual hemos realizado el proceso de mejora, para ser más específicos en el equipo de control de acceso vehicular.

¹ (POS) Siglas en inglés para Point of sale lo cual traduce terminal punto de venta (Conesa & Contreras, 2013)

Como se ha mencionado en un apartado anterior este equipo se encuentra integrado por una serie de dispositivos electrónicos, los cuales se interconectan a una controladora la cual es el núcleo de la operación de la máquina, la controladora es una placa compuesta por diferentes componentes electrónicos pero que es gobernada en su totalidad por la programación desarrollada en los microprocesadores presentes en ella. Los microprocesadores integrados en esta controladora por medio del programa desarrollado en la compañía, es el encargado de interconectar los diferentes elementos para realizar las funciones asignadas al expedidor de tarjetas, nombre con el cual se identifica el control de ingreso vehicular, los microprocesadores no solamente se encargan de gobernar la funcionalidad del equipo, también permite establecer la comunicación con los diferentes softwares que se han de implementar.

La controladora AccesScan, en su versión de desarrollo 4.1 fue diseñada durante el año 2003, en su diseño se implementó el microcontrolador Atmel AT89C51RD2, el cual pese a ser un dispositivo de alto uso a nivel industrial ya es por decirlo de cierta manera obsoleto, la antigüedad de este componente a la hora de querer realizar la integración de nuevas funcionalidades dentro del expedidor ocasiona grandes problemas de operatividad debido a que sus funciones operativas ya han sido copadas y carece de algunas funciones indispensables para interconectar nuevas tecnologías de manera óptima y sencilla, también encontramos otro problema, el cual no se relaciona con la parte referente al hardware o los componentes de la controladora, su principal problemática se encuentra en la forma que está construido el software o "Firmware" como se denomina el programa desarrollado para el control de la tarjeta. Este programa se encuentra desarrollado en C y quizás por el tiempo

con el cual tiene de implementación la persona que desarrollo el programa no aplico metodologías para contar con un código limpio, modular y escalable.

Si bien este tipo de deficiencias no son nuevas en la estructuración del programa, ¿por qué se continuo llevando a cabo su implementación dentro de los equipos distribuidos por la compañía durante todo este tiempo?, se podrían encontrar diferentes causas para dar respuesta a este interrogante, pero quizás la respuesta que nos daría cualquiera de las personas que estaban frente a estos proyectos en su momento seria, “ así ha funcionado desde un principio y no ha presentado ningún tipo de inconveniente”, puede que no sea una respuesta del todo certera y precisa, pero sin duda alguna seria la respuesta que cualquier persona que desconozca los problemas que se pueden generar por un cambio por más pequeño que sea, pero también por el hecho de que la persona que inicio el proyecto era la misma que continuaba dando soporte a la operatividad del sistema, realizando los cambios que eran requeridos o haciendo los ajustes al programa que fueran necesarios para mantener la controladora operando pese a los diferentes problemas que se pudieran presentar. Este último factor mencionado es quizás el que más peso tiene a la hora de poder determinar por qué no se habían realizado cambios en la forma de construcción del programa, o el por qué no era evidente encontrar un patrón de programación estructurada, con un programa escrito en código espagueti, con el uso de variables globales, con funciones sub utilizadas y la dependencia de respuesta dentro de los distintos ciclos implementados. Todos estos problemas estructurales fueron evidentes en el momento que una nueva persona tomo el programa y empezó a ver su estructura. Cuando fue necesario realizar el cambio del mecanismo de transporte y grabación de tarjetas inteligentes del expedidor por que la

compañía que lo fabricaba decidió discontinuar esta referencia dejando de dar cualquier tipo de soporte o venta de repuestos para soportar los equipos que aún se encuentran en funcionamiento, el simple cambio de este transporte requirió de muchas horas de análisis, de pruebas de operación, de entregas a clientes de sistemas operando a media marcha e inclusive con fallas operativas en algunas ocasiones recurrentes y notorias para el usuario final del sistema.

Es por esto que se ha desarrollado un nuevo firmware desarrollado bajo el sistema de lenguaje C en el cual se han implementado diferentes técnicas de arquitectura de software, creando clases independientes para los diferentes procesos de validación, lectura y grabación de las tarjetas inteligentes, del desacople de diferentes funciones dentro del sistema, además de esto se busca implementar en los procesos de desarrollo de la compañía, la implementación de buenas prácticas en programación y desarrollo de proyectos, lo que permitirá a la compañía ofrecer productos de calidad que cumplan con los estándares requeridos por los clientes de la compañía.

El contar con un firmware que permita el desacople de sus funciones y permita ser integrable con otras familias de microcontroladores permitirá realizar el cambio a corto de plazo de la controladora AccesScan.

1.3 Síntesis

El contar con un programa estructurado, basado en buenas prácticas de programación y siguiendo las pautas de las metodologías para la ejecución de proyectos, permitirá a la compañía Security Access FB disminuir los tiempos de integración de nuevas tecnologías dentro de la operatividad de los expedidores de tarjetas. Al contar con un programa que no

tenga dependencia de un transporte de tarjetas en particular se facilitará el proceso de reemplazo del tipo de mecanismo como también facilitara la integración de nuevos dispositivos electrónicos dentro de las controladoras a implementar a futuro.

2. Objetivo General

Realizar la reingeniería en el firmware de la controladora AccesScan para la implementación del transporte Lintech F3, mediante la implementación de patrones de diseño y de desarrollo de código limpio que faciliten la transición entre diferentes tecnologías de microprocesadores que permitirán agilizar el diseño de la nueva versión de la controladora fabricada y distribuida por la compañía Security Access FB.

3. Objetivos Específicos

- Estructurar de manera correcta las funciones operativas de la controladora.
- Crear librerías de operación independientes para los distintos dispositivos implementados en el expedidor de tarjetas.
- Habilitar la configuración de las funciones operativas de la controladora al personal técnico de la compañía.
- Restringir el acceso de usuarios no autorizados al módulo de configuración de la controladora mediante la generación de contraseñas aleatorias.
- Optimizar las tareas de lectura y grabación realizadas por el transporte de tarjetas inteligentes.
- Restructurar el proceso de ejecución de proyectos en el área de desarrollo de la compañía.

4. Marco de Referencia

4.1 Estado del arte

En los últimos años se han visto un crecimiento a gran escala de fabricantes de sistemas de control de acceso en diferentes mercados, especialmente por parte de los grandes fabricantes de sistemas de video, los cuales ya no solo ofrecen sistemas de video vigilancia, también ofrecen un amplio portafolio mediante el cual buscan satisfacer las necesidades de cualquier tipo sector económico o comercial, estas empresas a nivel general ofrecen sistemas muy completos, pero no tienen en cuenta las necesidades del mercado nacional, de los pequeños operadores y por supuesto a unos costos económicos elevados en la gran mayoría de los escenarios.

A nivel local en la última década han surgido compañías las cuales iniciaron siendo representantes de algunas de las grandes compañías extranjeras, pero con el paso de los años se han vuelto integradores o diseñadores de sus propios sistemas y productos, buscando llegar a poder satisfacer estas necesidades nacionales. Empresas que al igual que Security Access FB desarrollan sistemas para el cobro, auditoria, administración y control del ingreso a estacionamientos, también facilitando herramientas que permitan agilizar la recaudación de dinero por diferentes conceptos, el cobro por el uso de plazas de estacionamiento frecuente o extemporáneo, el acceso a servicios o muchos otros.

Dentro de los diferentes sistemas e integradores que se encuentran a nivel nacional una de las características que hace resaltar los ofrecidos por la compañía Security Access FB es el costo de integración de los dispositivos y la opción de mantenimiento o reparación que se brinda sobre cada una de las controladoras instaladas en los equipos, empresas nacionales

como EGLOBALT ofrecen equipos con operatividad similar a la ofrecida por nuestra compañía, con la gran diferencia que ellos en sus diferentes soluciones integran PLC's. De acuerdo con la definición de la NEMA un PLC es un aparato electrónico operado digitalmente el cual es programado para realizar funciones específicas a través de módulos de entrada/salida digitales o analógicos (Murillo, 2013). Si bien son soluciones de alta calidad, son mucho más costosas que una controladora fabricadas y diseñadas en nuestras instalaciones.

Otra característica importante es la posibilidad de brindar soporte y reparación de las controladoras que presenten cualquier tipo de falla funcional, labor que no es posible realizar en un PLC convencional o en las tarjetas de control ofrecidas por muchos de los distribuidores.

Aunque nuestras tarjetas de control compiten por costos y prestaciones operativas con otros sistemas, no podemos desconocer que la tecnología usada no es la mejor si es comparada con los controles de acceso importados, los cuales cuentan con dispositivos de control y almacenamiento de última tecnología los cuales permiten integrar diferentes opciones operativas dentro de sus sistemas. Por esta razón es evidente la necesidad de poder contar con un sistema que permita realizar cambios en sus funciones operativas de manera rápida sin afectar la integración con los programas de control y recaudo diseñados por la compañía.

En el mercado de las empresas internacionales y con un reconocimiento de alto desempeño podemos mencionar empresas tales como:

- Hikvision
- Dahua

- Axis
- Suprema
- Apolo
- HID
- FAAC
- CAME
- KeriSystems
- Bosch

Por nombrar solo algunas de las quizás más conocidas dentro del ambiente de los prestadores de servicios de seguridad y control a nivel global. La gran mayoría de estas compañías, si es que no son todas, a nivel nacional solo cuentan con empresas representantes para su instalación o distribución, pero en cuanto a reparación o integración de soluciones no ofrecen ningún tipo de servicio técnico o profesional.

Como se mencionó en un principio los grandes fabricantes de sistemas de video vigilancia han llegado al mercado nacional ofreciendo diferentes servicios y equipos para dar cubrimiento a todo lo referente sistemas de seguridad y acceso peatonal, también en cierta medida a los sistemas de control vehicular, pero no han centrado esfuerzo alguno en cubrir los aspectos operativos de la tarificación y recaudo de los dineros por lo cual buscan integradores a nivel local quienes puedan brindar este tipo de soluciones y brindar así un paquete integral, aspecto donde la operativa interna de nuestra compañía ha enfilado las nuevas propuestas comerciales y técnicas buscando crecer mediante el respaldo de grandes firmas internacionales.

Hace algunos años, la compañía en conjunto con el anterior ingeniero de desarrollo quisieron lanzar una mejor y más robusta línea de controladoras, en la cual eliminaban el uso de los dos microcontroladores, los cuales han sido uno de los grandes problemas a la hora de querer potenciar el uso de la controladora versión 4, pese a que esta controladora ya

era un producto maduro en muchos aspectos, recopilaba el proceso de todos los años de desarrollo ejecutados, integraba diferentes equipos para la impresión y el manejo de transportes de tarjetas, seguía presentando las fallas de modularidad en el sistema, los problemas de comunicación con los transportes de tarjetas continuaban generando bloqueo en las operaciones de comunicación y recepción de comandos entre los diferentes sistemas. De esta nueva controladora se alcanzaron a instalar algunas unidades, pero por sus repetidos problemas de comunicación y sobre todo por el bloqueo presentado en su módulo de comunicación TCP-IP debió ser remplazada, retornando a la versión anterior, la cual, pese a los problemas ya mencionados a la hora de querer escalar nuevas tecnologías, sigue soportando la totalidad de los sistemas instalados.

Los acontecimientos presentados desde el segundo trimestre de este año 2020 a nivel mundial con la aun presente emergencia sanitaria ha expuesto las debilidades organizativas que se tienen en diferentes sectores empresariales, especialmente pequeñas y medianas empresas, donde no se cuenta con controles o seguimientos a los procesos desarrollados en ellas.

Los sectores económicos deben tener la posibilidad de adaptarse a cualquier tipo de cambio de manera oportuna y eficiente, esta reciente emergencia nos ha enseñado que depender de un solo nicho comercial puede ser peligroso al momento de presentarse cualquier cambio en la manera como se asume la economía de una organización, es por esto y por las distintas deficiencias operativas de los sistemas actuales de la compañía Security Access FB que se busca implementar la reestructuración en el departamento de desarrollo de la

compañía, partiendo desde uno de los implementos que ofrece el mayor de los soportes a los equipos de la compañía.

4.2 Impacto

El contar con un programa estructurado, basado en buenas prácticas de programación y ejecución de proyectos, permitirá a la compañía Security Access FB disminuir los tiempos de integración de nuevas tecnologías dentro de la operatividad de los expedidores de tarjetas, al igual que la integración de nuevas funciones a los procesos ejecutados por la controladora. Al contar con un programa que no tenga dependencia de un transporte de tarjetas en particular se facilitará el proceso de reemplazo del tipo de mecanismo e inclusive cambiar la tecnología, cambiando los mecanismos de transporte de tarjetas por sistemas de impresión térmica.

El desarrollo de nuevos módulos en el programa de la controladora sin duda agilizará los procesos de configuración de las funcionalidades operativas de los equipos de control de acceso al personal técnico de la compañía, funcionalidad protegida por el sistema de encriptación de llaves de acceso.

4.3 Componente de innovación

El proceso de reingeniería realizado en el programa de la controladora AccesScan, permitirá contar con un software único para la el manejo del dispensador de tarjetas Lintech F3, esta funcionalidad se obtendrá al contar con la posibilidad de realizar las labores de configuración de los modos de operación de las controladoras directamente en el dispositivo eliminando la necesidad de crear un software para cada una de las controladoras a instalar en las diferentes máquinas de control de acceso vehicular, esta mejora permitirá

reducir el tiempo dedicado por parte del personal de desarrollo de la compañía en el levantamiento de los requerimientos de cada instalación, del proceso de generación de los programas, de la realización de pruebas de operatividad y del posterior seguimiento de operación del sistema.

Otra de las ventajas que se obtendrá de contar con este tipo de funcionalidad es el poder tener un repositorio de fácil acceso y comprensión para todo el personal técnico de la compañía que requiera grabar el programa en cualquiera de las controladoras sin importar su ubicación geográfica o la hora de ser requerido, todo esto protegido por la validación de doble factor que se ha implementado en el sistema para realizar la configuración de la controladora, para este fin se desarrolló una App que facilitara el generar las claves de acceso al sistema de configuración, eliminando la necesidad de contar con una computadora para validar el acceso al módulo de configuración.

Un factor adicional de gran importancia dentro del proceso realizado es el de contar con un programa escalable para el uso de otras familias de microprocesadores, habilitando la posibilidad de realizar a corto plazo el diseño y desarrollo de una nueva versión de controladora la cual ofrezca mayores funciones operativas.

4.4 Marco teórico

La automatización de procesos es algo que toma más fuerza y crece a medida que lo hacen las tecnologías implícitas en su aplicación, por ello hoy en día es común encontrar diferentes propuestas comerciales con una gran variedad de productos y con precios cada vez más accesibles, si bien los procesos de automatización surgieron como una solución aplicada a

procesos industriales, cada día toman más fuerza en diferentes campos, tales como la aplicación en tareas del hogar, de funciones corporativas entre otras.

Dependiendo el nivel de complejidad de nuestra tarea o función a automatizar podremos encontrar a nivel comercial soluciones sencillas, las cuales nos permiten realizar acciones tales como; encender o apagar bombillos, abrir o cerrar cortinas, asegurar puertas y muchas otras. También encontraremos propuestas mucho más robustas las cuales se ejecutan mediante el uso de dispositivos especializados, dispositivos lógicos programables (Dispositivos lógicos programables - EcuRed, s. f.), los cuales permiten el manejo de múltiples señales de entrada y salida, diseñados para soportar condiciones extremas de uso, capaces de operar muchas horas seguidas con bajo nivel de mantenibilidad, los fabricantes de estos dispositivos, dependiendo de sus características operativas y funcionales nos brindan todo tipo de artefactos a diferentes precios, desde el económico de combate hasta la gama de lujo.

Estas tecnologías desarrolladas por compañías extranjeras, nos limitan al momento de tener que realizar labores de reparación o adaptación de operatividad bajo ciertas características. Por esta y quizás otras razones, la compañía desde sus inicios se encamino en brindar soluciones desarrolladas a nivel nacional, partiendo del diseño y fabricación de dispositivos que permitieran controlar funciones operativas de equipos en un mercado definido, estacionamientos y accesos peatonales. La búsqueda de brindar una solución integral llevo a buscar procesos desde el ensamble de las tarjetas electrónicas al desarrollo de los programas específicos para el control de los dispositivos conectados a estas.

Los equipos son funcionales hoy en día, pero muchos de estos dependen de dispositivos ensamblados por grandes fabricantes, los cuales con el pasar de los años catalogan ciertas referencias de productos como obsoletos terminando con su fabricación y distribución a nivel mundial, lo cual sucedió en los últimos años, la compañía recibió un comunicado por parte del fabricante de los transportes de tarjetas que la referencia usada en los equipos de la compañía, saldrían de circulación por completo, esto obligo la implementación de un nuevo transporte en los equipos ofrecidos nuestra compañía, sumado al cambio del personal encargado de las labores de desarrollo de los sistemas, genero grandes inconvenientes en el proceso de migración hacia el uso de los nuevos transportes mecánicos que se empezarían a implementar. Uno de los grandes retos encontrados por el nuevo equipo de desarrollo se evidencio al notar que los sistemas hasta ese momento soportados dependían en gran medida de ciertos comandos de respuesta brindados por el antiguo transporte y que la falta de estas respuestas afectaba por completo la funcionalidad del sistema, también se encontró que el código de programación heredado, no contaba con un protocolo definido, su estructura estaba conformada por numerosas interacciones anidadas, también se encontraron variables no usadas y otras globales, las cuales en cualquier momento podrían presentar fallas en alguno de los procesos del ciclo operativo del sistema.

Razones como las mencionadas generaron grandes traumatismos en los procesos de entrega de nuevos proyectos, también en la solución a requerimientos de actualizaciones o mejoras solicitadas por parte de los clientes ya existentes, las cuales para ser realizadas requirieron de horas e inclusive de meses de estudio, de diferentes ajustes y gran cantidad de pruebas,

para poder brindar una solución que cumpliera con los requerimientos solicitados y cubriera en cierta medida las peticiones de los clientes finales.

Hemos referenciado uno de los dispositivos fabricados por la compañía Security Access FB, el cual hace parte de una de las soluciones ofrecidas para facilitar el control del personal que ingresa a centros comerciales o complejos empresariales, para nuestro caso, el control de acceso de vehículos a estacionamientos. Por supuesto este tipo de control se puede implementar sistemas de acceso peatonales para controlar el uso de puertas, torniquetes o exclusas e inclusive en el manejo de sistema de seguridad controlando el accionamiento de señales de alerta o disparadores externos, pero en nuestro caso puntual lo centramos en el manejo exclusivo del primer escenario, es decir, el control de acceso de vehículos a estacionamientos. Partiendo de esto describiremos a continuación cuales son los elementos que conforman un control de ingreso vehicular como también el modo de operación del mismo;

- Tarjeta controladora AccesScan; tarjeta electrónica diseñada y ensamblada en la compañía, esta controladora cuenta con dos microprocesadores los cuales son el núcleo de la operatividad de la misma, la controladora permite la conexión de mecanismos mediante comunicación RS232 y sus funciones operativas son gobernadas por señales externas, figura 1. La controladora adicionalmente se encarga de realizar la conexión con el software de control.



Figura 1. Controladora AccessScan versión 4. Autoría propia.

- Transporte de tarjetas Lintech F3; este mecanismo se encarga de transportar las tarjetas MIFARE de un compartimiento de almacenamiento para realizar la validación y grabación de la información pertinente al registro de ingreso para ser entregada al cliente o visitante, también puede capturar tarjetas desde el exterior, esta operación corresponde al manejo de abonados o mensuales, figura 2.

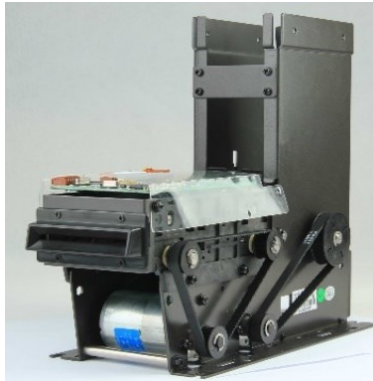


Figura 2. Transporte de tarjetas inteligentes Lintech F3. Tomada de (Automatic Card Dispenser Parking Lot System Dispenser, s. f.)

- Pantalla de usuario; se pueden contar con diferentes tipos de pantallas según la versión del quipo, figura 3, estas pantallas permiten obtener información del estado de operatividad al visitante, como también de brindar información mediante mensajes informativos del proceso de ingreso a la instalación.



Figura 3. Pantallas informativas del expedidor de tarjeta, se presentan tres de los modelos adaptables. Autoría propia.

- Lector de tarjetas de proximidad; estos lectores permiten la lectura de tarjetas de uso exclusivo a personal autorizado, en la mayoría de los casos mensualidades o usuarios VIP que realizan uso del estacionamiento, este tipo de lectura es funcional únicamente si la controladora tiene comunicación total con el software, figura 4.
- Sensor de presencia vehicular; el sensor permite activar las funciones de la controladora una vez es detectado el vehículo en la zona designada.



Figura 4. Lector de tarjetas Mifare ZKTeco. Adaptada de (ZKTECO PROID10BM, s. f.)

- Pulsador; permite la entrega de tarjetas a usuarios que son visitantes en las instalaciones.

Los dispositivos mencionados son los componentes básicos de cualquier sistema de control de ingreso vehicular ofrecido por la compañía Security Access FB en la actualidad, adicional a esto existen una serie de funciones auxiliares las cuales son integrables al equipo, estas funciones son controladas por medio de un programa externo de la compañía, el cual mediante una serie de comandos enviados por la controladora AccesScan, realiza las funciones requeridas, estas funciones son:

- Digitalización de placas; permite capturar mediante una serie de funciones externas los datos de las placas de vehículos que ingresan a las instalaciones.
- Inventario; registro fotográfico mediante la conexión de cámaras de video, permite almacenar el estado del vehículo al momento de ingresar a las instalaciones.

Hasta este punto no se ha descrito con detalles los procesos realizados al momento de realizar el ingreso de un vehículo a un estacionamiento, pero ya se cuenta con una idea general de los diferentes componentes y sus funciones dentro del expedidor de tarjetas, también describimos las dos funciones opcionales dentro de un sistema de ingreso vehicular.

En este proyecto se han utilizado las herramientas Keil uVision5 y Android Studio 4.0 en lo referente a entornos de desarrollo, cada uno de estos entornos de desarrollo han sido utilizados por sus características funcionales y operativas al momento de la ejecución de las soluciones implementadas y por supuesto por sus características funcionales y operativas. Keil uVision5 es un entorno de programación que permite la generación de código en los lenguajes C y C++, diseñado esencialmente el manejo de microcontroladores que permite la generación de software embebido en dispositivos basados en ARM Cortex-M, incluye herramientas para la depuración de los programas desarrollados. Este entorno ayuda a acelerar el proceso de desarrollo al proporcionar diferentes herramientas compatibles con más de 7500 dispositivos.

Android Studio es por su parte el entorno de desarrollo oficial para la plataforma Android, gracias a su potente editor de código y a las diferentes herramientas que proporciona este

entorno de desarrollo, se obtiene una mayor productividad en el desarrollo de aplicaciones apps para Android.

5. Metodología

En el desarrollo del proyecto implementaremos el uso de las metodologías de programación extrema XP apoyada por el manejo metodología Kanban, la selección de la metodología extrema XP es debido a sus características funcionales y modelo de implementación, también por el tipo de interacción del proyecto y por la facilidad de integrar algunas de sus prácticas a lo largo del proceso.

De esta manera cuanto más cambios, puede que más cerca estemos del mejor resultado que espera nuestro cliente, este cambio constante en el proyecto se llega a considerar como favorable (Canive & Balet, s. f.). Esta metodología nos permite tomar las fortalezas de los integrantes del grupo, por una parte, la amplia experiencia en desarrollo de firmware de un ingeniero electrónico, y el amplio conocimiento de diferentes tipos de controladoras como también de equipos de control de acceso peatonal y vehicular de diferentes distribuidores e integradores a nivel nacional de un ingeniero en control, lo cual es una de las bases de implementación de esta metodología, la programación en parejas.

Otro punto a favor de esta metodología es su forma de interacción con el cliente, dado que, en muchos de los proyectos de este tipo, las entregas al cliente se realizan de forma parcial, solucionando requerimientos esenciales, llegando a una etapa de satisfacción total. En cuanto a la implementación como apoyo en el proceso de la metodología Kanban se debe a la facilidad de contar con un sistema visible para los avances y los procesos que se deben tener en cuenta por parte del equipo de desarrollo, también teniendo presente que en el

proceso de reingeniería llevado a cabo se busca mejorar el comportamiento de un sistema que se encuentra funcional y el apoyo del personal técnico de la compañía entra a formar parte fundamental dentro del proceso de mejora llevado a cabo.

Para la implementación de esta metodología en primera instancia se planificará con el cliente, en nuestro caso el gerente de la compañía, el alcance que se le dará al proyecto.

Posterior a la planificación se realizará un estudio del diseño actual y se plantearan soluciones a implementar en el sistema, para este diseño se contempla contar con personal externo al proyecto, como es el caso de un director técnico quien conoce las fallas operativas del sistema actual. Contando con los conocimientos generales del estado del sistema actual se iniciará con la codificación de la aplicación, esta se realizará por segmentos de operación, partiendo de un sistema de ingreso básico en el cual solo se grabará la información pertinente a un registro común y finalizando con un sistema que permita contar con la validación de usuarios autorizados mediante la lectura de tarjetas ingresadas al transporte. Cada una de las modificaciones será acompañada por una serie de pruebas funcionales, ya que la compañía nos brindará los recursos necesarios para esta implementación.

6. Proceso de Software

6.1 Requerimientos funcionales

Definidos los alcances por parte de los diferentes actores involucrados dentro de la compañía Security Access FB en el proceso de reingeniería desarrollado para el firmware del microcontrolador se detectaron los siguientes requisitos funcionales en el sistema.

- Garantizar la comunicación con los softwares actuales de la compañía:

Todo proceso de validación o expedición de tarjetas deberá ser enviado a los sistemas de control de la compañía, AccesScan y Monitor.

- Conectividad con los dispositivos externos del expedidor:

El nuevo firmware debe mantener la operatividad de los equipos actualmente utilizados en los diferentes modelos de equipos distribuidos por la compañía, se debe garantizar que las pantallas de información, los mecanismos de lectura de tarjetas de proximidad y los elementos de detección vehicular continúen siendo operables en caso de requerir una actualización en las instalaciones ya operativas.

- Manejo de mensualidades o autorizados mediante los métodos ya establecidos en el mercado actual:

El nuevo firmware podrá ser compatible con los dos modelos de operación de mensualidades del sistema AccesScan, proximidad de tarjetas o lectura mediante inserción en el transporte de tarjetas. Las funciones deben ser operativas en las dos modalidades logrando mantener las funciones actuales del sistema.

- Configuración de modo operativo de la controladora:

Se deberá poder realizar ajustes en el modo de operación de la controladora de manera que se puedan realizar acciones operativas según los requerimientos del cliente.

- Operatividad autónoma:

La controladora podrá continuar su operación habitual en caso de presentarse desconexión de los sistemas de software de la compañía. La controladora deberá operar sin ningún inconveniente así no se tenga respuesta alguna de los sistemas de

control de acceso instalados en los equipos de cómputo del cliente final, la controladora debe ser autónoma para garantizar el funcionamiento de los ingresos vehiculares.

- Usabilidad de la controladora AccesScan versión 4:

El firmware desarrollado debe garantizar la funcionalidad de la controladora a pesar de las características técnicas de sus componentes actuales.

6.2 Requerimientos no funcionales

Los requerimientos no funcionales tenidos en cuenta son.

- Garantizar el acceso al sistema de configuración de la controladora.
- Mejorar los tiempos de grabación de la tarjeta inteligente por parte del transporte mecanizado.
- Migrar funciones operativas no soportadas por los sistemas actuales de la compañía, eliminar la integración con sistemas no utilizados en los actuales proyectos soportados por la compañía.
- Implementación de módulos de configurables, la controladora deberá ser configurable sin importar el modelo a seguir.
- Programa de acceso a la configuración de la controladora con interfaz gráfica o segmentada para su manejo.
- Despliegue de programa generador de llaves para equipos móviles con sistema IOS.

6.3 Diseño y arquitectura

El proceso de reingeniería implementado en el programa del microprocesador secundario de la controladora AccesScan, permitirá contar con un sistema de calidad. Mediante la eliminación de problemas de estructura y funcionalidad, como también de las malas prácticas de desarrollo presentes en el código heredado.

Aunque el planteamiento inicial del proyecto solo buscaba implementar un nuevo transporte de tarjetas, durante el desarrollo del proyecto, se logró la implementación de un módulo de configuración funcional con todas las características operativas del antiguo sistema, todo esto protegido mediante la generación de claves de seguridad.

6.3.1 Diagrama de despliegue

A continuación, se describe el despliegue de los procesos desarrollados conforme a las funciones realizadas por la controladora AccesScan, ver figura 5.

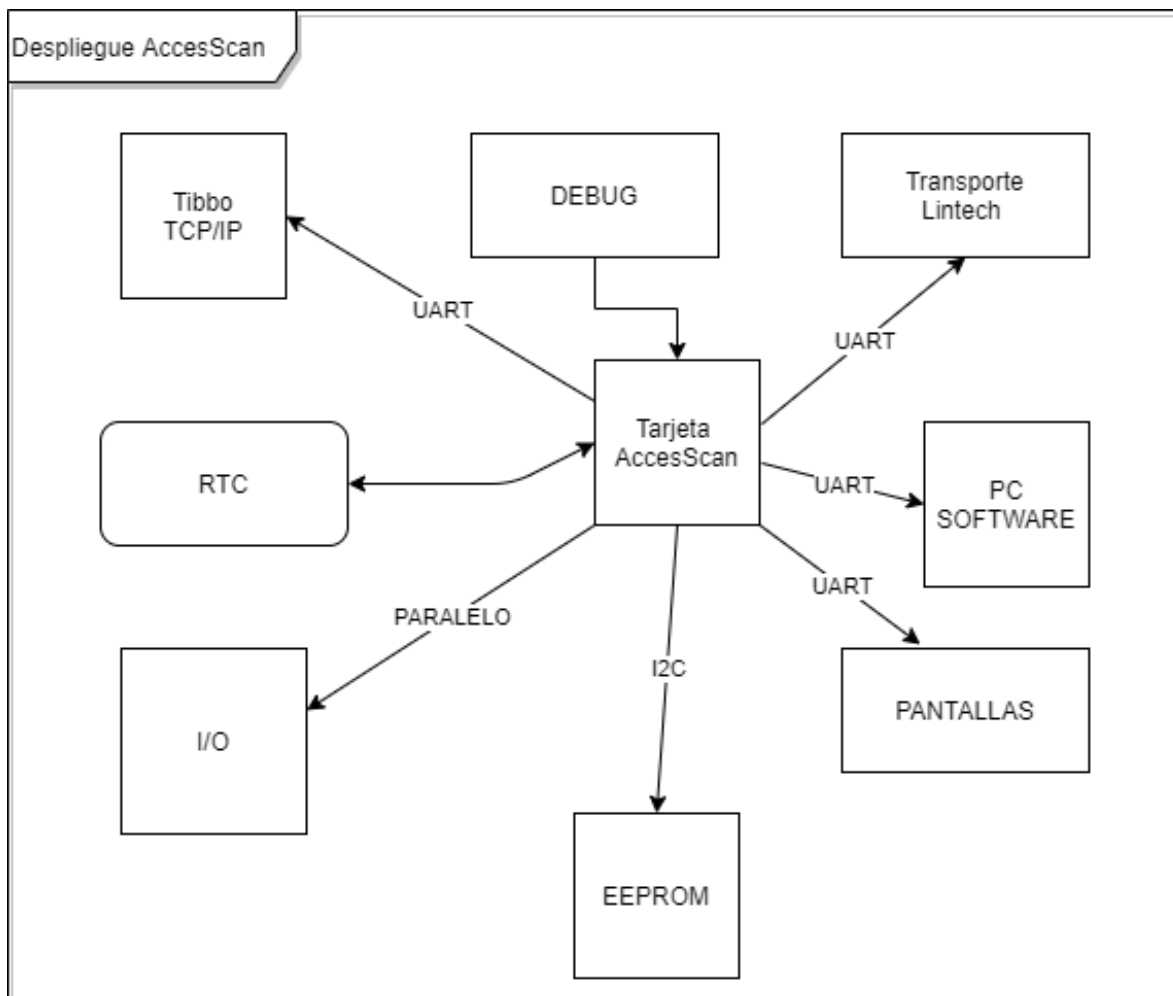


Figura 5. Diagrama de despliegue, conexión de la controladora con los diferentes equipos y sistemas integrados en los expedidores de tarjetas inteligentes. Autoría propia

6.3.2 Caso de uso arquitectura relevante

El caso de uso general del sistema se presenta en la figura 6, este caso de uso nos permite apreciar los dos actores que interactúan con la controladora de manera directa o indirecta, el primero de ellos es identificado como visitante y corresponde a toda persona que hará uso de los equipos de ingreso en las instalaciones, independiente de su categoría. El segundo de los actores corresponde al técnico de la compañía Security Access FB, quien será la persona encargada de realizar el proceso de adecuación y puesta en funcionamiento del

sistema. El detalle del caso de uso podrá encontrarse en el anexo *A. Aplicación de metodología.*

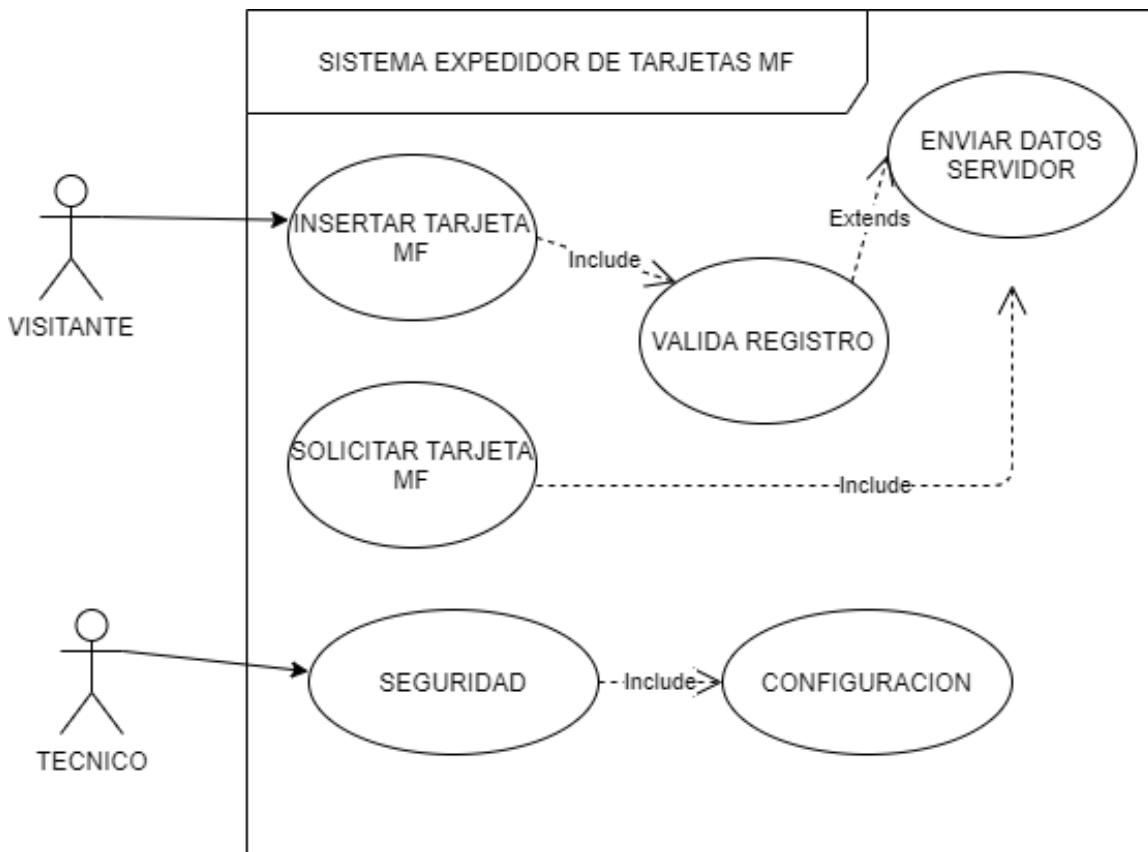


Figura 6. Diagrama de caso de uso de la operación normal del sistema. Autoría propia.

6.3.3 Diagrama de secuencia

El desarrollo del firmware para el microcontrolador se divide en dos grandes procesos, la expedición de tarjetas para los visitantes y el manejo de tarjetas autorizadas o mensualidades. Para el primer actor nos podemos remitir a la figura 7, en este diagrama de secuencia se pueden identificar los pasos funcionales desarrollados por la controladora.

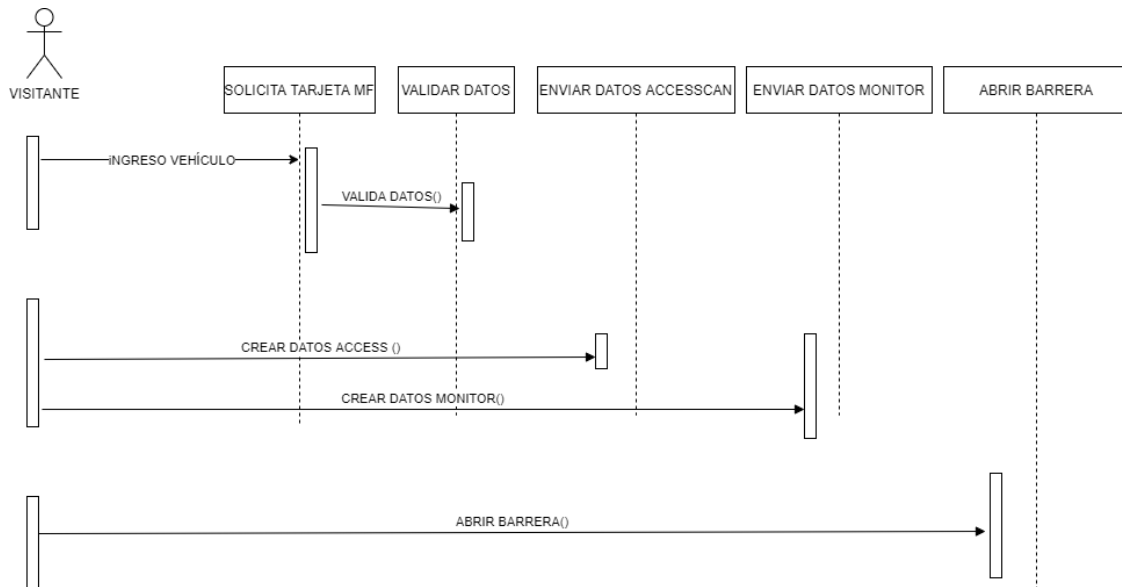


Figura 7. Diagrama de secuencia para el manejo de tarjetas de visitante o rotación en el sistema. Autoría propia.

En cuanto a la funcionalidad del sistema durante el proceso de ingreso de los mensuales, podemos detallar las secuencias de operación en la figura 8.

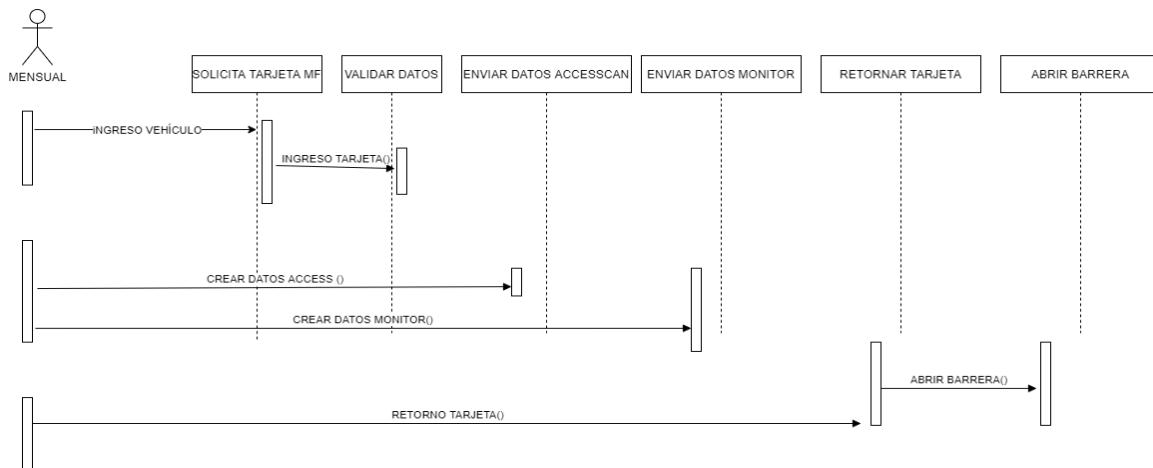


Figura 8. Diagrama de secuencia manejo de tarjetas de proximidad con ingreso por transporte de tarjetas. Autoría propia.

6.3.4 Arquitectura de alto nivel

El sistema descrito corresponde al control de acceso vehicular mediante el uso de expedidores de tarjetas inteligentes o tarjetas MIFARE, la implementación de estos sistemas era encontrado principalmente en sitios con alto flujo vehicular, aunque hoy en día es común encontrar estas soluciones en cualquier ámbito comercial. La figura 9 nos permite apreciar el ciclo regular de operación de un sistema de control de acceso a grandes rasgos.

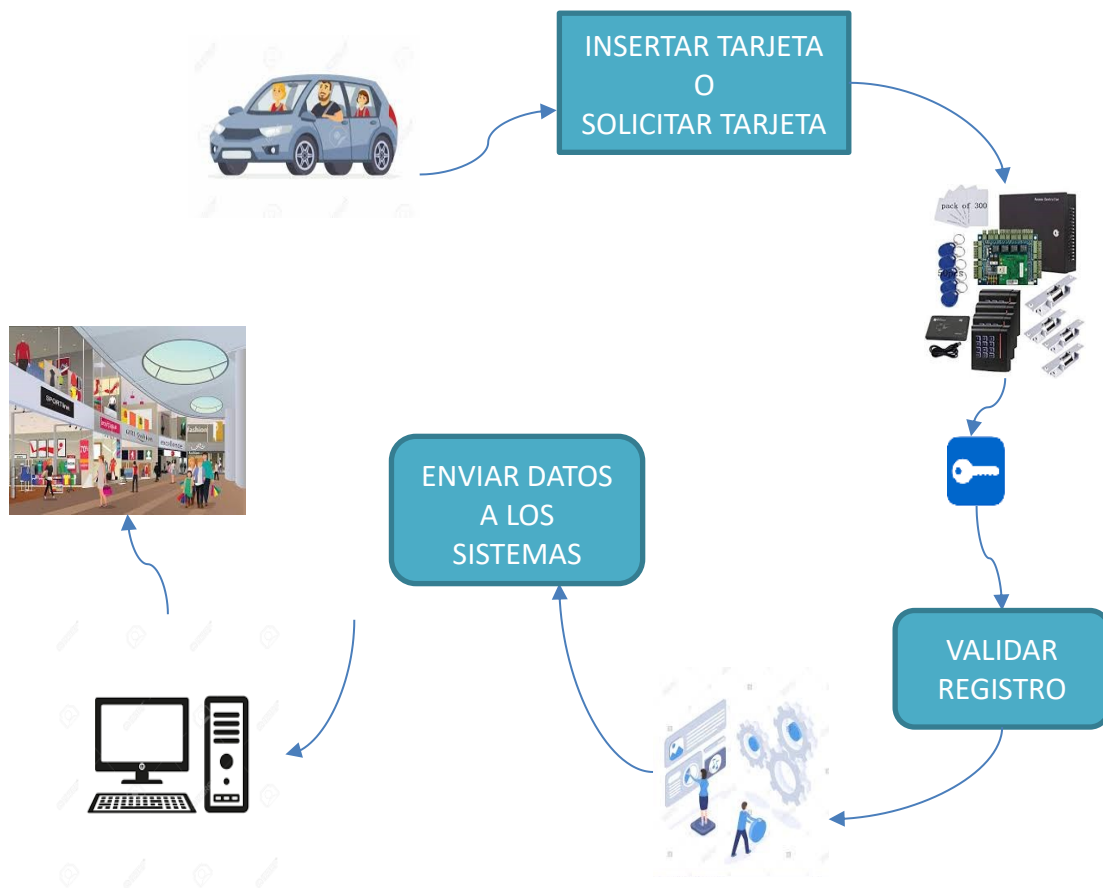


Figura 9. Proceso de registro de ingresos al sistema de control vehicular mediante la expedición de tarjetas operado por la controladora AccesScan. Autoría propia.

6.4 Construcción

El proceso implementado en la reingeniería del código del microcontrolador secundario de la controladora AccesScan surge de la necesidad de mejorar los tiempos de entrega de las

modificaciones solicitadas por los diferentes clientes de la compañía Security Access FB y principalmente por lo complejo de mantener el código. Pese a esto la organización es conforme con tener código que funcione, que compile o que tenga contenido al cliente, sin darle importancia a la calidad del mismo. Hemos hecho un esfuerzo para que la organización sea consciente de la deuda técnica que estamos llevando.

El programa está desarrollado siguiendo el estándar ANSI C, el cual formaliza construcciones para trabajar con sistemas embebidos (microcontroladores / microprocesadores). Para lograr buenos resultados en este proceso se realizó un análisis del anterior código encontrando que:

- Se contaba con una modelo de programación secuencial o código espagueti “clases o funciones extremadamente largas del orden de cientos de línea de código, con innumerables sentencias `if` o con muchos bloques anidados”(Blanes, s. f.). Lo cual deja de ser un patrón pasando a convertirse en un problema. En la figura 10 evidenciamos una secuencia de un **case SEQ_RD_S1B2_M** que tiene 519 líneas de código secuenciales, lo que puede provocar que alguien que no lo conozca o que no lo trabaje habitualmente, si debe modificarlo no va a tener muy claro que es lo que está haciendo. Esta es una de las malas prácticas encontradas en el código heredado.

```

10310     case SEQ_RD_S1B2_M:
10311         if ((ValTimeOutCom==1) || (bandera_rx_soft==1))
10312         {
10313             bandera_rx_soft=0;
10314             if (g_cContByteRx==21) //NUMERO DE DATOS = 21
10315             {
10316             }
10317             else
10318             {
10319
10320                 send_portERR(ERR_LECTURA);
10321                 led_err_imp=0;
10322                 eject_hold();
10323                 g_cEstadoComSeqMF=SEQ_EJECT_INICIO; // MESULA VA A LO NEVOOOO
10324                 g_cEstadoComSoft=ESPERA_RX;
10325                 ValTimeOutCom=20;
10326             }
10327         }
10328
10329         break;

```

Figura 10. Código Secuencial. Secuencia de lectura de uno de los bloques de la tarjeta Mifare. Autoría propia.

- Otra desventaja evidente es la gran cantidad de variables globales usadas en este programa lo que hace difícil de entenderlo mantener, depurarlo y mejorarlo. Esto se puede evidenciar al encontrar 293 variables globales y 83 booleanos “testigos o bits”, ver figura 11.

46	unsigned int	CicloTimeAudio=0;	124	bit	info=0;
47	unsigned char	cte_seg;	125	bit	info_ticket=0;
48	unsigned char	seg;	126	bit	compara_data=0;
49	unsigned char	seg2;	127	bit	viene_err=0;
50	unsigned char	byte;	128	bit	viene_stacker=0;
51	unsigned char	num_bytes;	129	bit	toggle=0;
52	unsigned char	dir_h;	130	bit	proceso_iniciado=0;
53	unsigned char	dir_l;	131	bit	once=0;
54	unsigned char	dir_clk;	132	bit	once2=0;
55	unsigned char	byte_out;	133	bit	passA=0;
56	unsigned char	num_datos;	134	bit	passB=0;
57	unsigned char	nbitsW;	135	bit	onceClotofono=0;
58	unsigned char	cod_err;	136	bit	onceMsg=0;
59	unsigned int	ValTimeOutCom;	137	bit	refresh_clk=0;
60	unsigned char	num digitos;	138	bit	SendTrigger=0;
61	unsigned char	NumChrLPR=0;	139	bit	SendTriggerB=0;
62	unsigned char	num_datos;	140	bit	StatusSend=0;
63	unsigned char	TimeCmdRaspberry=0;	141	bit	Dir_MotoCarro=0;
64	unsigned int	t_espera;	142	bit	Espera_RtaCMD=0;
65	unsigned char	Trigger;	143	bit	Time_OutW=0;
66	unsigned char	year_max;	144	bit	Disabled_AB=0;
67	unsigned char	mes_max;	145	bit	Valido=0;
68	unsigned char	dia_max;	146	bit	InfoNo_Card=0;
69	unsigned char	hora_max;	147	bit	Debug_Port_Pantalla;
70	unsigned char	min_max;	148	bit	sendactive=0;
71	unsigned long int	SALDO;	149	bit	Explico=0;
			150	bit	NotifyJam=0;
			151	bit	Eject_Card=0;
			152	bit	Accesso_Permitted=0;
			153	bit	Fuera_Lineas=1;
			154	bit	TimeOut=0;
			155	bit	EnQueue=0;
			156	bit	Fuera_Horario=0;
			157	bit	Dia_FicopIaca=0;
			158	bit	Habilidad_FicopIaca=0;
			159	bit	AccesoOtorgado=0;
			160	bit	Placa_In=0;
			161	bit	PrintLPR=0;
			162	bit	SendCard_Date;
			163	bit	W...=0;

Figura 11. Declaración de variables y bits. Muestra de la gran cantidad de variables declaradas en el código anterior. Autoría propia.

- El acoplamiento del programa con diferentes funciones es alto lo que hace que el firmware haga diferentes tareas y el programa no se pueda modularizar. Un ejemplo de ello es la función de grabación de los datos en las tarjetas MIFARE, no es una sola función de escritura, también valida y hace conversiones. Para escribir en otra página de la tarjeta MIFARE el código es duplicado y así se repite la misma función varias veces, no hay reutilización de código lo cual aumenta la deuda técnica en el proyecto, lo cual se ve reflejado en un esfuerzo extra en el desarrollo o la codificación de funciones, todo esto aumenta por consiguiente el tiempo de desarrollo, la duplicidad de código se puede apreciar en la figura 12.

```

3554 //*****
3555 void graba_S1B2_out(void) //EJEMPLO 02 00 13 39 06 01 0C 07 09 10 14 00 00 00
3556
3557 {
3558     unsigned char bcc,j;
3559
3560     bcc=0;
3561     //-----
3562     g_sckrrTxComSoft[0]=STX; //STX
3563     g_sckrrTxComSoft[1]=0X00; //ADDRESS
3564     g_sckrrTxComSoft[2]=0X13; //LEN
3565     g_sckrrTxComSoft[3]=0X39; //CMD
3566     g_sckrrTxComSoft[4]=0X06; //SECTOR 1 BLOQUE 0 = 0X06
3567     g_sckrrTxComSoft[5]=0X01; //
3568     //-----
3569     g_sckrrTxComSoft[6]=0; //g_sckrrRxComSoft[4];
3570     g_sckrrTxComSoft[7]=0; //g_sckrrRxComSoft[5];
3571     g_sckrrTxComSoft[8]=0; //g_sckrrRxComSoft[6];
3572     g_sckrrTxComSoft[9]=0; //g_sckrrRxComSoft[7];
3573     g_sckrrTxComSoft[10]=0; //g_sckrrRxComSoft[8];
3574
3575     //-----
3576     // EJECUCION
3577     //-----
3578     // Borra Descuento y pone APB
3579     //***** DESCUENTO
3580     g_sckrrTxComSoft[11]=0x00; //g_sckrrRxComSoft[9];
3581     g_sckrrTxComSoft[12]=0x00; //g_sckrrRxComSoft[10];
3582     g_sckrrTxComSoft[13]=0x00; //g_sckrrRxComSoft[11];
3583     //----- TIPO VEHICULO
3584     if (Tipo_Tarjeta==ROTACION)
3585     {
3586         g_sckrrTxComSoft[14]=TipoVehiculoIN; //0X00;
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621 //*****
3622 void graba_S2B0(void) //EJEMPLO 02 00 13 39 06 01 0C 07 09 10 14 00 00
3623
3624 {
3625     unsigned char bcc,j;
3626
3627     bcc=0;
3628     //-----
3629
3630     g_sckrrTxComSoft[0]=STX; //STX
3631     g_sckrrTxComSoft[1]=0X00; //ADDRESS
3632     g_sckrrTxComSoft[2]=0X13; //LEN
3633     g_sckrrTxComSoft[3]=0X39; //CMD
3634     g_sckrrTxComSoft[4]=0X08; //SECTOR 1 BLOQUE 2 = 0X08
3635     g_sckrrTxComSoft[5]=0X01; //
3636     //-----
3637     temp=6;
3638     if ((NumChrLPR<=10) && (Placa_In==1))
3639     {
3640         for (j=0; j<NumChrLPR-2; j++) // Quita 2 caracteres < >
3641         {
3642             g_sckrrTxComSoft[temp++]=buffer_placa[j+1];
3643         }
3644         for (j=temp; j<22; j++)
3645         {
3646             g_sckrrTxComSoft[temp++]=0x00;
3647         }
3648     }
3649     g_sckrrTxComSoft[14]=buffer_PLATE_PREPAY[8];
3650     g_sckrrTxComSoft[15]=buffer_PLATE_PREPAY[9];
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703

```

Figura 12. Funciones duplicadas. Detalle del uso de la misma función para realizar la misma acción de grabación en las tarjetas Mifare. Autoría propia.

- El firmware no posee una forma de parametrización, para diferentes tipos de clientes y los parámetros que tiene son booleanos esto crea una complejidad ciclomática “métrica del software ampliamente difundida y aceptada, que permite obtener una medición cuantitativa de la complejidad lógica de un programa o pieza de software.”(Cardacci, 2016,p.2). El código está basado en numerosos caminos independientes, que lo hace difícil de entender, mantener y probar.
- El control de versiones no es el adecuado los cambios requeridos se traducen en una nueva versión y una nueva carpeta del firmware teniendo el día de hoy más de 200 versiones con diferentes funciones operacionales, para cada uno de los clientes se requiere de la generación de una carpeta contenedora, figura 13. En este punto no se tiene presente cual es la función de cada uno de los firmwares almacenados.

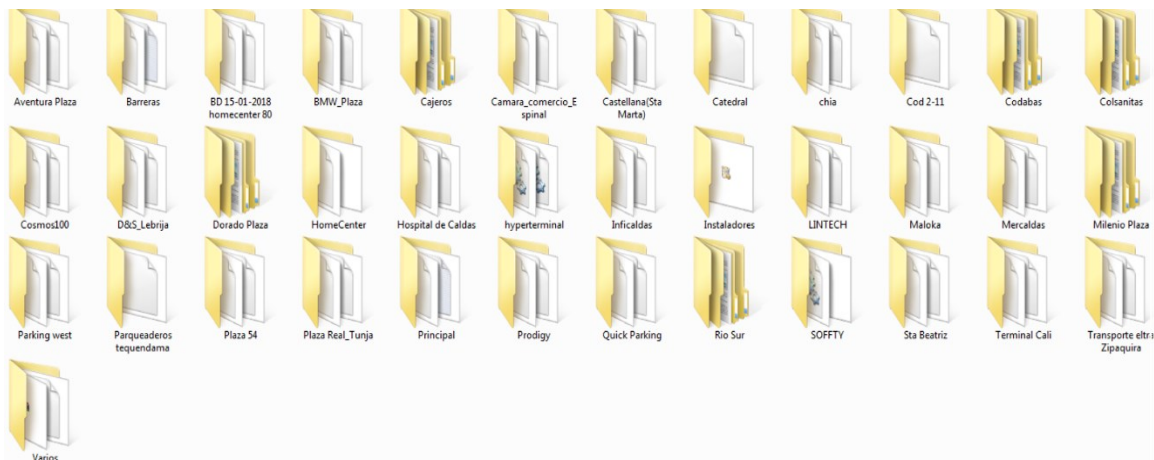


Figura 13. Control actual de versiones. Cada carpeta corresponde a un solo cliente, en este se pueden encontrar más de un firmware almacenado. Autoría propia.

- La tarjeta controladora tiene varios componentes asociados los cuales deben tener un firmware para comunicarse entre ellos. Este software no cuenta con módulos

independientes para los diferentes componentes de hardware, todo está en un solo archivo, ver figura 14.

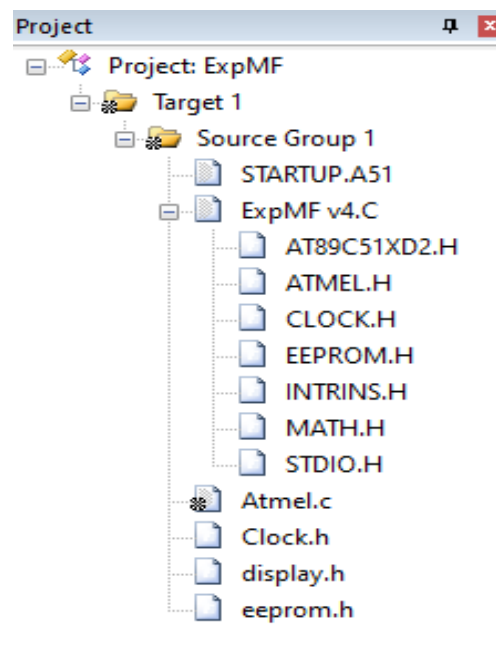


Figura 14. Árbol del proyecto, estructura firmware anterior. Autoría propia.

En la reingeniería del firmware con los pocos recursos que disponemos en el microcontrolador usado en la controladora AccesScan, hemos implementado lo aprendido en asignaturas como; Arquitectura, Calidad de software y Seguridad, seleccionando las herramientas más adecuadas, según las necesidades y expectativas del sistema. Con el fin de llegar a escribir código cada vez más profesional y acorde con estándares de calidad, seguridad y confiabilidad, para sistemas embebidos se debe conocer y dominar gradualmente las idiosincrasias de cada uno de los patrones o métodos de programación

más usados. Se ha implementado una máquina de estados finita (MEF)². En el cual podemos abstraer el número de estados existentes en el sistema y cada estado realizara un numero definido de acciones, o mostrar un comportamiento específico a este estado. Como los estados son conocidos de antemano, un determinado grupo de eventos causaran cambios de un estado a otro de acuerdo a reglas específicas. Cada tarea es realizada con un tiempo de secuencia.

La manera más usual de implementar las MEF en lenguaje C es mediante bloques switch – case, esto se aprecia en la figura 15.

```

unsigned char SecuenciaExpedidorMF( unsigned char EstadoActivo)
{
    static unsigned char Buffer_Write_MF[17];
    static unsigned char Atributos_Expedidor[15];
    static unsigned char Secuencia_Expedidor[4];
    static unsigned char Nombre_Mensual[17];

    switch (EstadoActivo)
    {
        //*****
        case SEQ_INICIO:
            if ((ValTimeOutCom==True) || (ValTimeOutCom > TIME_CARD))
            {
                break;
            }
        case SEQ_CMD_ACEPTADO:
            /*cmd comun para todos*/
            EstadoActivo=rta_cmd_transporte(Secuencia_Expedidor);
            break;
        case SEQ_MOVER_CARD_RF:
            Mov_Card(MovPos_RF);
            EstadoActivo=Load_Secuencia_Expedidor (Secuencia_Expedidor,EstadoActivo,SEQ_CMD_ACEPTADO,SEQ_TIPO_CARD);
            break;
        case SEQ_CARD_INSERTION_ON:
            Card_Insertion(Habilita);
            EstadoActivo=Load_Secuencia_Expedidor (Secuencia_Expedidor,EstadoActivo,SEQ_CMD_ACEPTADO,SEQ_RESPUESTA_TRANSPORTE);
            Secuencia_Expedidor[TareadelCmd ] = TAREA_TIPO_MENSUAL;
            break;
        case SEQ_TIPO_CARD:
            Aut_Card_check_Status();
            EstadoActivo=Load_Secuencia_Expedidor (Secuencia_Expedidor,EstadoActivo,SEQ_CMD_ACEPTADO,SEQ_RESPUESTA_TRANSPORTE);
            Secuencia_Expedidor[TareadelCmd ] = TAREA_TIPO_TARJETA;
            break;
        case SEQ_LOAD_PASSWORD:
            LoadVerify_EEPROM();
            EstadoActivo=Load_Secuencia_Expedidor (Secuencia_Expedidor,EstadoActivo,SEQ_CMD_ACEPTADO,SEQ_READ_SECTOR_BLOQUE);
            Atributos_Expedidor [ Sector ] = Sector_1;
            Atributos_Expedidor [ Bloque ] = Bloque_1;
            break;
        case SEQ_READ_SECTOR_BLOQUE:
            RD_MF(Atributos_Expedidor [ Sector ],Atributos_Expedidor [ Bloque ]);
            EstadoActivo=Load_Secuencia_Expedidor (Secuencia_Expedidor,EstadoActivo,SEQ_CMD_ACEPTADO,SEQ_RESPUESTA_TRANSPORTE);
            Secuencia_Expedidor[TareadelCmd ] = TAREA_LECTURA_TARJETA_SECTOR_BLOQUE;
    }
}

```

Figura 15. Implementación de máquinas de estado finito en el nuevo firmware de la controladora. Autoría propia.

² (MEF) Máquina de estados finito

En cierto modo cada estado “guardado en memoria”, es una instancia de eventos y estados pasados. Esta característica nos ayuda a reducir las variables globales y banderas en la aplicación. En la figura 16 se muestran los estados y tareas del programa.

```

132 /
133 Definiciones de secuencias de verificador y expedidor
134 -----
135 #define SEQ_INICIO                               0X00
136 #define SEQ_RESPUESTA_TRANSPORTE               0X01
137 #define SEQ_CMD_ACEPTADO                     0X02
138 #define SEQ_MOVER_CARD_RF                      0X03
139 #define SEQ_CARD_INSERTION_ON                  0X04
140 #define SEQ_TIPO_CARD                           0X05
141 #define SEQ_LOAD_PASSWORD                      0X06
142 #define SEQ_READ_SECTOR_BLOQUE                 0X07
143 #define SEQ_WRITE_SECTOR_BLOQUE               0X08
144 #define SEQ_CAPTURE_CARD                       0X09
145 #define SEQ_CARD_INSERTION_OFF                 0X0a
146 #define SEQ_EXPULSAR_CARD                      0X0b
147 #define SEQ_LOAD_EEPROM                       0X0c
148 #define SEQ_FRONT_CARD                        0X0d
149 #define SEQ_ESPERA_VEHICULO_ENTRE              0X0e
150 #define SEQ_DETAIL_CARD                        0X0f
151 #define SEQ_PTO_PARALELO                       0X10
152 #define SEQ_WAIT_PLACA                         0X11
153 #define SEQ_UID                               0X12
154 #define SEQ_LPR                               0X13
155 #define SEQ_TIPO_TARJETAS                      0X14
156 #define SEQ_CAPTURE_CARD_LOOP                 0X15
157 #define SEQ_POWER_OFF                          0X16
158 #define SEQ_POWER_ON                          0X17
159
160 /*-----
161 Definiciones de secuencias de tareas del verificador y expedidor
162 -----
163 #define TAREA_PRESENCIA_VEHICULAR              0X00
164 #define TAREA_TIPO_TARJETA                     0X01
165 #define TAREA_LECTURA_TARJETA_SECTOR_BLOQUE  0X02
166 #define TAREA_WRITE_TARJETA_SECTOR_BLOQUE    0X03
167 #define TAREA_OPEN_BARRERA                     0X04
168 #define TAREA_WRITE_PLACA_CARD                0X05
169 #define TAREA_TIPO_MENSUAL                     0X06
170 #define TAREA_UID                             0X07
171 #define TAREA_PRESENCIA_ROTACION              0X08

```

Figura 16. Definición de los estados y tareas en el nuevo firmware de la controladora. Autoría propia.

Un modelo representativo de la MEF implementada en el microcontrolador y que se encarga de gestionar las funciones realizadas por la controladora se puede apreciar en la figura 17.

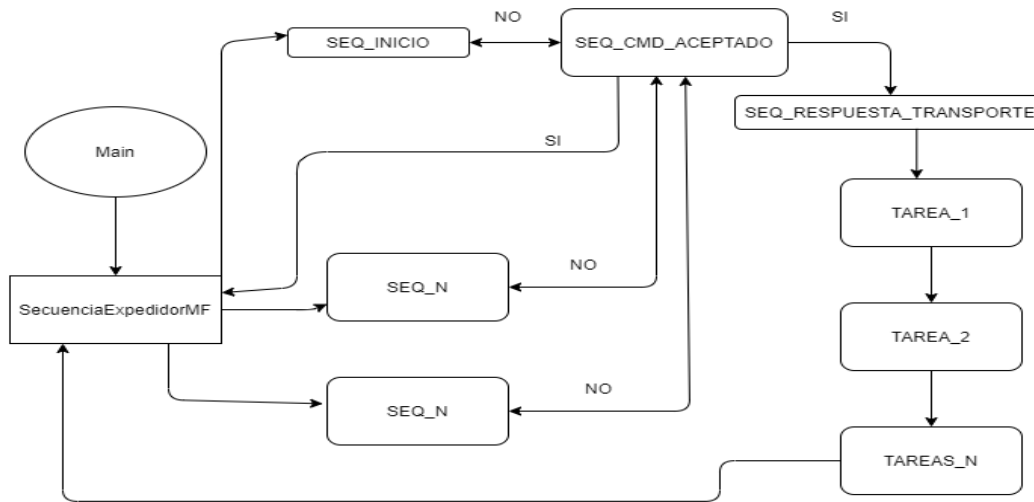


Figura 18. Representación de las máquinas de estado finito. Autoría propia.

La gran ventaja de la implementación de las MEF en el firmware de la controladora AccesScan se evidencio en la reducción de variables globales, en el nuevo código es solo se declararon catorce variables globales y los Booleanos usados son tan solo dos. La figura 18 nos permite apreciar la diferencia de variables declaradas entre los dos firmwares.

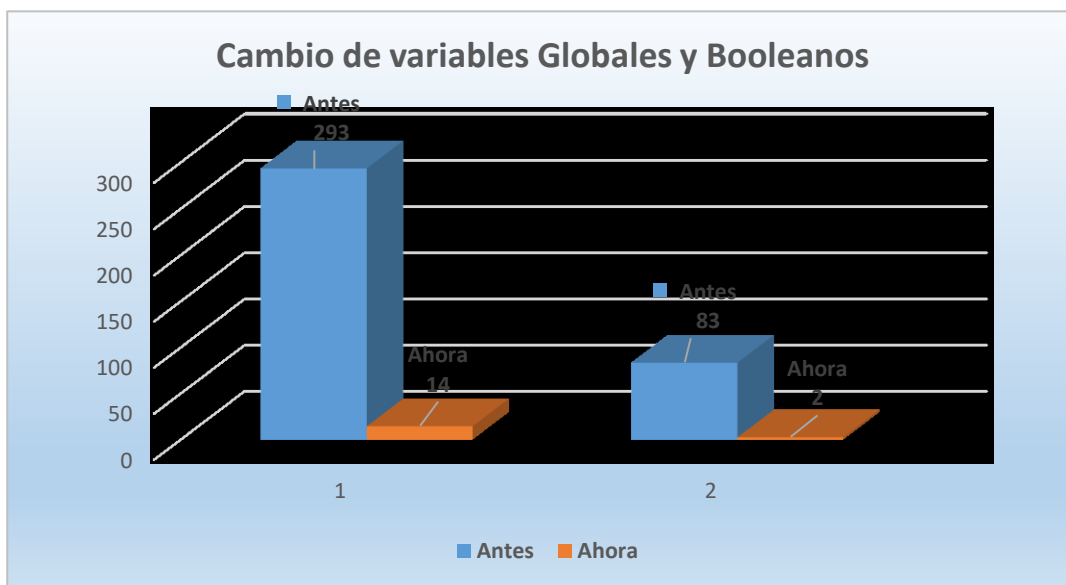


Figura 17. Comparativa de la declaración de variables en los firmwares, antes y después del proceso de reingeniería. Autoría propia.

Mediante la creación de paquetes definidos, se logró obtener el desacople presente en código reemplazado. El hardware está dividido en varios componentes los cuales tienen tareas y funcionalidades independientes que fueron segmentadas en diferentes módulos, al realizar esta mejora en el programa se pudo facilitar la comprensión de este permitiendo ofrecer una mayor mantenibilidad del desarrollo implementado, en la figura 19 se pueden apreciar los diferentes dispositivos interconectados a la tarjeta electrónica.

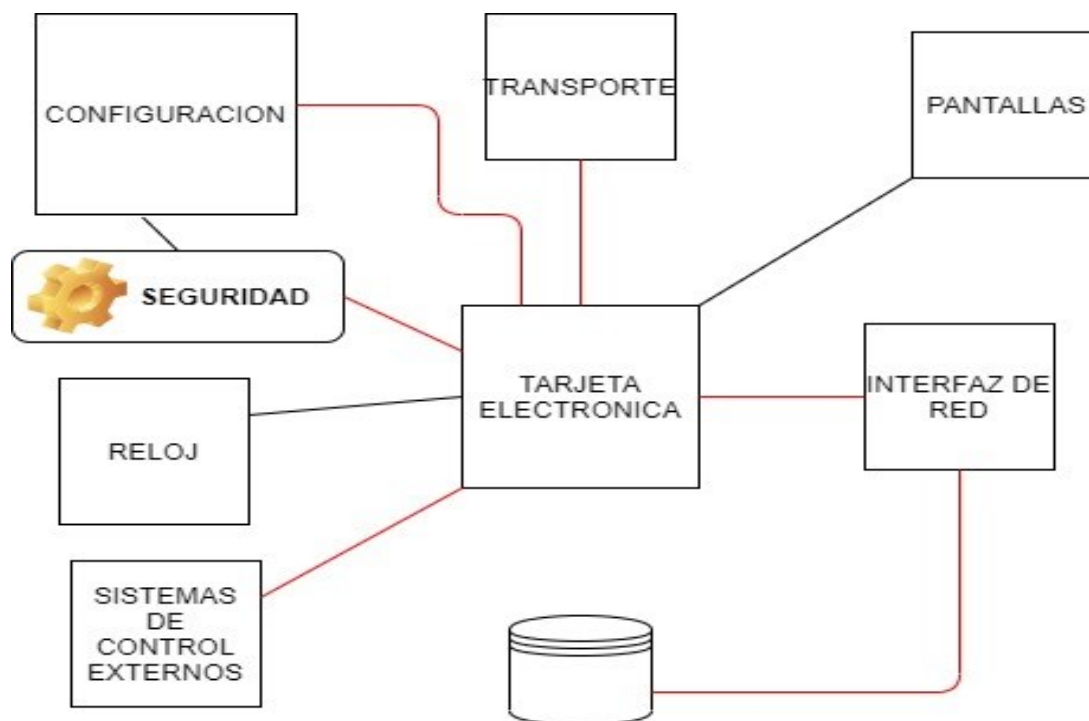


Figura 19. Dispositivos presentes en la controladora. Autoría propia.

La modulación del código ha sido estandarizada mediante la implementación de un árbol de componentes donde cada uno de estos es asociado por medio del hardware o software y

donde cada uno de estos es totalmente independiente el uno del otro, con funciones desacopladas y con una alta cohesión en la funcionalidad que debe cumplir. Figura 20

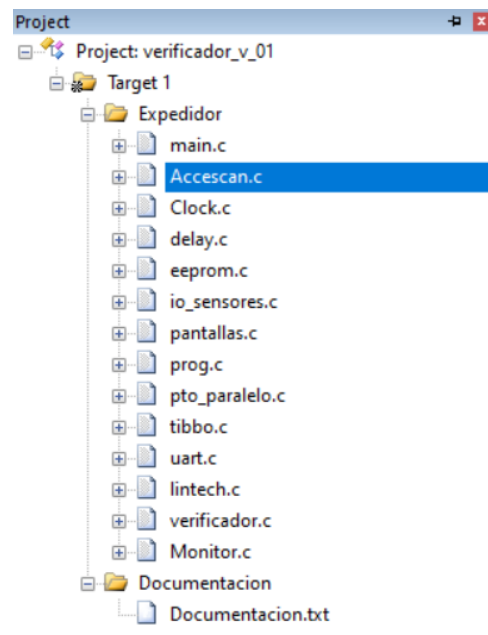


Figura 20. Módulos implementados en el proyecto. Autoría propia

6.5 Pruebas

Este proceso ha sido un camino lleno de dificultades y es cuando uno como ingeniero de desarrollo debe sacar su ingenio, para probar el código ya que las pruebas unitarias de cada función son más complejas porque el software va empotrado en algo físico que es el microcontrolador. Entonces ideamos un sistema de pruebas basado en un log por TCP/IP mediante el cual podemos conocer el estado en el que se encuentra el programa y nos permite ver los resultados de los procedimientos y funciones ejecutados, en ese momento nosotros podemos validar la respuesta y saber si es el resultado es correcto o mirar si se

queda en algún bug, o podemos aislar una parte del programa lo cual nos facilita la localización del error.

Este sistema ha sido de gran ayuda ya que mediante este se han logrado encontrar fallas presentadas en diferentes procesos entre las cuales se encuentran algunas que llevaban años tratándose de solucionar, con la solución de algunas de estas fallas se ha podido cumplir con un mínimo de calidad en el firmware dando solución a los problemas del sistema.

Tenemos otras pruebas que son las funcionales donde simulamos situaciones de estrés para romper la aplicación, en estas el firmware debe validar esos eventos inesperados donde se le colocan otros tipos de tarjetas que no debe reconocer o tarjetas no validas, ante estas situaciones el transporte funcionara conforme a los parámetros establecidos en el firmware de la controladora y tomara las decisiones correspondientes, además se han realizado pruebas de carga, simulando momentos de uso masivo buscando alterar la estabilidad de los equipos, todas esta pruebas son monitorizadas por el debugger de la controladora.

Mediante el uso del debug del sistema fue posible encontrar fallas presentadas en las instalaciones que no son detectadas durante las pruebas realizadas en el departamento de desarrollo, las pruebas son realizadas con equipos bajo operaciones de funcionamiento completamente diferentes a las presentadas en una instalación normal, especialmente si estas instalaciones son actualizaciones, donde los equipos de cómputo y el sistema de comunicaciones pueden tener años de uso.

Gracias al proceso de seguimiento que realiza el debug en la controladora se validan los pasos realizados por el sistema permitiendo realizar modificaciones en la operación del

sistema o contemplar cambios futuros en el modo de operación de la controladora, en la figura 21 se puede apreciar el proceso de un seguimiento realizado.

```

TAREA_PRESENCIA_VEHICULAR<CR><LF>
Nivel bajo de tarjetas<CR><LF>
Sensores Auto y sensor1 activos.<LF><CR>
Pulsador Activo<CR><LF>
Moviendo Tarjeta a RF<CR><LF>
Datos Enviados al Transporte: F2 00 00 03 43 32 32 03 B1 <CR><LF>

26/10/2020 16:22:43.224 [RX] - Datos Recibidos del Transporte: F2 00 00 06 50 32 32 31 30 03 94 <CR><LF>
Aut_Card_check_Status<CR><LF>
Datos Enviados al Transporte: F2 00 00 03 43 50 31 03 D0 <CR><LF>
Datos Recibidos del Transporte: F2 00 00 08 50 50 31 32 31 30 31 30 03 FA <CR><LF>
TAREA_TIPO DE TARJETA<CR><LF>
Tarjeta valida MF50<CR><LF>
Carga y Verifica de EEprom<CR><LF>
Datos Enviados al Transporte: F2 00 00 07 43 60 33 00 21 00 01 03 C6 <CR><LF>

26/10/2020 16:22:44.555 [RX] - Datos Recibidos del Transporte: F2 00 00 08 50 60 33 32 31 30 90 00 03 59 <
Leyendo MF > Sector: 01 Bloque: 01 <CR><LF>
Datos Enviados al Transporte: F2 00 00 08 43 60 33 00 B0 01 01 01 03 58 <CR><LF>
Datos Recibidos del Transporte: F2 00 00 18 50 60 33 32 31 30 01 05 00 20 00 00 00 00 00 00 00 00 00
TAREA_LECTURA_TARJETA_SECTOR1_BLOQUE1<CR><LF>
Buffer_s1_b1<CR><LF>
Datos Recibidos del Transporte: 01 05 00 20 00 00 00 00 00 00 00 00 00 00 00 00 <CR><LF>
CARD ID CLIENTE: 032<CR><LF>
ID CLIENTE: 032<CR><LF>
CARD COD_PARK:005<CR><LF>
COD_PARK:005<CR><LF>
TIPO DE TARJETA: 01 <CR><LF>
Leyendo MF > Sector: 01 Bloque: 02 <CR><LF>
Datos Enviados al Transporte: F2 00 00 08 43 60 33 00 B0 01 02 01 03 5B <CR><LF>
Datos Recibidos del Transporte: F2 00 00 18 50 60 33 32 31 30 00 00 00 00 00 00 00 00 00 03 02 00 00 00 00
TAREA_LECTURA_TARJETA_SECTOR1_BLOQUE2<CR><LF>
Buffer_s1_b2<CR><LF>
Datos Recibidos del Transporte: 00 00 00 00 00 00 00 00 00 03 02 00 00 00 00 00 <CR><LF>
HORARIO:00 <CR><LF>
PICO Y PLACA:00 <CR><LF>
ANTIPASSBACK:02 <CR><LF>
TIPO DE TARJETA ROTACION<CR><LF>
<LF><CR>
/*-----*/<LF><CR>
Recibe trama pto paral= Datos Recibidos del Transporte: B6 <CR><LF>
<LF><CR>
longitud de la trama: 01 <LF><CR>
<LF><CR>
/*-----*/<LF><CR>

```

Figura 21. Segmento de debug realizado en los procesos de pruebas y seguimiento.

Autoría propia.

Durante el proceso de inicio del proyecto se evidenció que los firmware generados no contaban con un sistema que permitiera realizar un seguimiento de versiones o cambios realizados, para esto basándonos en los nuevos conocimientos adquiridos se implementó el uso de herramientas que nos permitieran gestionar y controlar los cambios realizados en el código desarrollado, se crearon dos repositorios, uno local con Git y otro remoto privado con GitHub, ver figura 22. En estos repositorios se almacenaron los diferentes archivos generados durante la ejecución del proyecto mediante la ejecución de commits. Un commit

se relaciona con la acción de guardar o subir archivos a un repositorio remoto o local, una actualización de los cambios (Cuadrados, 2015).

The image shows a terminal window on the left and a GitHub commit history table on the right. The terminal window displays the output of the command `git log --oneline`, showing a list of commit hashes and their corresponding messages. The GitHub table shows the commit history for the repository, including the commit hash, the commit message, and the time since the commit was made.

Commit Hash	Commit Message	Time Since Commit
a92433d	Octubre 20 del 20 se mueve el estado respuesta transporte	hace 5 días
b4026b8	ajuste en tarjeta invalida, variable de estadotarificador se pas a estatica	hace 5 días
95aa409	expedidor tarjeta mf50 y mf70 prog clave sin *	hace 3 meses
95279e8	se organizo la libreria de msj	hace 6 meses
d0cd50f	expedidor con direccion diferente para monitor	hace 6 meses
ab91ce1	Version con arreglo trama pto paralelo	hace 6 meses
09f6f06	(tag: f985d07, tag: b7dade4, tag: 7784fe7, tag: 6e67524, tag: 64614e7, tag: 3c7afe9, tag: 1da1fe1, tag: 03) version funcional con prog completa	hace 6 meses
b7dade4	version 19_08_2020 ver programacion de horarios habilitados	hace 6 meses
7784fe7	version 19_08_2020 mejora prog comando adicional ver programacion	hace 6 meses
1da1fe1	version con msj agosto 4 2020	hace 3 meses
f985d07	version funcional expedidor con 2 msj de bienvenido faltantes	hace 3 meses
3c7afe9	version 14_07_2020	hace 6 meses
64614e7	msj version 21-05-20	hace 6 meses
6e67524	version inicial expedidor	hace 6 meses

Figura 22. Repositorio de versiones en Git y GitHub. Autoría propia.

La implementación de estas herramientas nos permitió tener claridad de los avances obtenidos en la aplicación, también nos facilitó el manejo del firmware creado ya que se mantendría el mismo nombre evitando el almacenamiento innecesario de archivos. Inicialmente se había estructurado realizar el despliegue de las versiones con GitHub y SonarCloud, buscando así que cada despliegue contara con una medición de calidad, desafortunadamente no se pudo depurar por el software de calidad ya que SonarCloud solo valida el IDE de programación de IAR y nosotros trabajamos con Keil uVision C51. Actualmente estamos validando la implementación de nuevos microprocesadores los cuales permiten la integración de nuevas herramientas de desarrollo para el compilador Keil uVision e inclusive son configurables mediante el uso de plugins en el software Eclipse, estas nuevas versiones de programas de desarrollo para sistemas embebidos permiten la integración del aplicativo SonarLint.

6.6 Instalación y configuración

Una de las mejoras que ha tenido mayor impacto dentro de la organización y consideramos que es de las más notoria fue hacer que el firmware con el hardware existente, sea adaptable a las necesidades del cliente, sin necesidad de la intervención del personal de desarrollo o de personal con un alto grado de conocimiento o un manejo de herramientas especializadas. Gracias a la creación de un sistema de parametrización totalmente programable por uno de los puertos seriales de la controladora donde están las necesidades conocidas en más de 7 años de desarrollo en aplicaciones de estacionamiento, han permitido que el personal de soporte técnico pueda hacer los ajustes de configuración en la instalación conforme a las necesidades del cliente.

Sin embargo, durante la adecuación de este módulo de configuración nos percatamos que el dejar esta nueva funcionalidad sin ningún tipo de seguridad sería causante de una brecha de seguridad dentro de la funcionalidad del sistema, ya que cualquier persona con un grado de conocimiento básico en el sistema o en manejo de software podría realizar modificaciones dentro de la funcionalidad operativa de los equipos.

Crear la seguridad para un microcontrolador es difícil por los tamaños del hash y nuestra gran limitante es la memoria para usar algoritmos conocidos y muy complejos. Entonces ideamos un algoritmo que fuera robusto el cual nos asegurara que con la respuesta nunca pueda saber cuáles son los datos insertados. Creamos nuestro sistema de doble autenticación donde la tarjeta genera una Id de autenticación de 10 caracteres alfanuméricos que pueden ser números letra mayúsculas y minúsculas, ver figura 23. Con este Id tenemos

dos aplicaciones que desarrollamos una en Android y otra en Java donde digitamos el Id generado por el hardware y el software me genera la nueva llave de registro.



Figura 23. Aplicaciones para la generación de las claves de registro. Autoría propia.

Esta llave de registro es enviada por el puerto serial y si es válida nos deja ingresar al módulo de programación. En la figura 24 se evidencia la autenticación del sistema.

```

/>Id Registro:zpPZzhzZhb<CR><LF>
<LF>
/>Password:
25/10/2020 13:50:31.113 [TX] - jgIgj3XZ3z<LF><CR>

25/10/2020 13:50:31.116 [RX] - jgIgj3XZ3z<LF><CR>
<CR><LF>
ID_CLIENTE      --- CMD 0 Identificador del cliente maximo 255<CR><LF>
COD_PARK        --- CMD 1 El numero del parqueadero maximo 255<CR><LF>
T_GRACIA        --- CMD 2 Tiempo sin cobro 00, maximo 255<CR><LF>
SIN_COBRO       --- CMD 3 Inhabilitado= 0, Gratis= 1 , Gratis y salida el mismo dia=
DEBUG           --- CMD 4 Habilitar = 1, Inhabilitar = 0<CR><LF>
USE_LPR         --- CMD 5 Habilitar = 1, Inhabilitar = 0<CR><LF>
TIPO_PANTALLA   --- CMD 6 PANTALLA LCD =0 PANTALLA RASPBERRI=1<CR><LF>
CARD_AUTOMATICA --- CMD 7 BOTTON=0 AUTOMATICA=1<CR><LF>
ANTIPASSBACK    --- CMD 8 Habilitar = 1, Inhabilitar = 0<CR><LF>
PLACA           --- CMD 9 Habilitar = 1, Inhabilitar = 0<CR><LF>
ADDRESS_HIGH_BOARD --- CMD 10 La direccion alta del board del numero 5 al 9 sino se usa
HORARIO         --- CMD 11 Progama 10 horarios del 1 al 10<CR><LF>
VALIDA_VEH_MENSUAL --- CMD 12 Habilitar = 1, Inhabilitar = 0<CR><LF>
USE_APB_MENSUAL --- CMD 13 Habilitar = 1, Inhabilitar = 0<CR><LF>
VER_PROGRAMACION --- CMD 14 Muestra la programacion<CR><LF>
AYUDA          --- CMD 15 Muestra los comandos<CR><LF>
SALIR          --- CMD 16 Salir de programacion<CR><LF>

```

Figura 24. Acceso al módulo de programación mediante la generación de llaves de acceso. Autoría propia.

En este punto se logró dar solución a nuestros problemas de acceso al módulo por parte de personal ajeno a la organización, otro de los riesgos contemplados es el retiro de personal técnico de la compañía, el cual puede quedarse con una copia de los programas de generación de llaves, para mitigar este nuevo posible riesgo, se acondiciono un método de vencimiento por fechas de las llaves generadas por estas aplicaciones, con la implementación de este vencimiento programado, si la persona deja de laborar en la compañía, los medios de generación de llaves serán obsoletos y no podrá ingresar a realizar modificaciones en la controladora.

7. Anexos

A. Aplicación de metodología

Pese a que la metodología extrema XP se centra la conformación de equipos de por lo menos 5 roles, para poder llegar a buen término en la implementación de esta en nuestro proyecto, nos hemos repartido los diferentes roles entre las dos personas que conforman el grupo de trabajo, de esta manera los roles han sido manejados como se indica en la tabla 1.

Tabla 1.
Roles en la metodología extrema XP

Rol	Actor interno
Cliente	Coordinador de desarrollo
Programador	Jaime Pedraza
Tester	Edwin Rodríguez
Coach	Jaime Pedraza
Manager	Edwin Rodríguez

La tabla 1 nos permite apreciar los diferentes roles cumplidos por los integrantes del grupo de trabajo. Autoría propia.

Definidos los roles dentro del proceso se procedió a realizar la primera reunión de manera virtual, debido a los acontecimientos presentados durante el transcurso del año, en esta se contó con el acompañamiento del gerente general de la compañía y el coordinador del área de desarrollo quien a su vez sería el cliente durante todo el proceso.

De esta reunión se lograron definir los alcances funcionales del sistema y los requerimientos mínimos para una primera etapa de pruebas, tabla 2.

Tabla 2
Primera iteración, inicio del proyecto.

Historia de Usuario	
Iteración: 1	
Ciente: Coordinador del área de desarrollo	
Nombre historia: Inicio de proyecto	
Prioridad: Alta	Riesgo: Baja
Programador responsable: Jaime Pedraza – Edwin Rodríguez	
Descripción:	
<p>El firmware de la controladora debe ser compatible con el nuevo transporte de tarjetas inteligente Lintech, para esto se debe conservar las funciones básicas de operación del sistema para mantener la correcta operación de los sistemas ante nuestros clientes actuales, para esto la controladora debe poder contar con la opción de entregar tarjetas de manera automática o por medio de la obturación del pulsador, también la controladora debe poder mantener la comunicación con el sistema AccesScan y Monitor con el fin de poder brindar las funciones administrativas de los sistemas. En esta primera etapa el ingreso de abonados o mensualidades se realizará mediante el manejo de lectores de proximidad.</p>	
Observaciones: Se contempla la falta de recursos físicos los cuales se deberán soportar lo antes posible.	
La tabla permite apreciar los procesos solicitados por el firmware en su etapa inicial.	

Autoría propia.

El caso de uso para este requerimiento es identificado como caso de uso inicial, figura 25.

En este se definen los dos modelos de operación para ingreso a las instalaciones, mensual o autorizado y visitante en donde el expedidor realizara las acciones conforme a:

- Mensual o autorizado: Se realizará el proceso de lectura de la tarjeta mediante el lector de proximidad, esta información se enviará al sistema AccesScan el cual validará el estado del registro para autorizar el ingreso del mensual, si el sistema no autoriza el ingreso se retornará un mensaje al expedidor informando al usuario porque no se permite el ingreso, si el desea continuar con el ingreso pulsará el botón para solicitar tarjeta como visitante.
- Visitante: El usuario pulsará el botón del equipo solicitando la tarjeta de ingreso, en esta tarjeta el transporte debe grabar toda la información pertinente al registro del ingreso, la información a grabar en la tarjeta será; consecutivo, fecha, hora, tipo de vehículo “Carro o moto”, placa del automotor (si cuenta con el sistema).

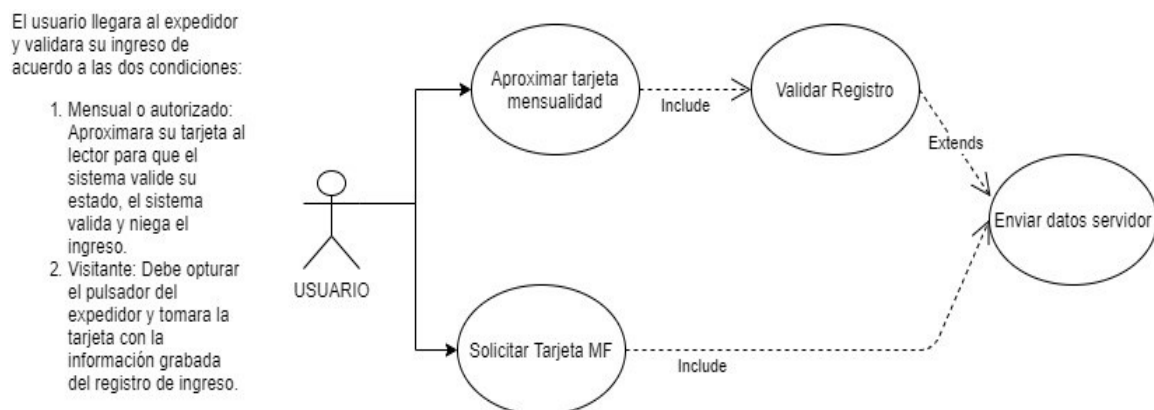


Figura 25. Primer caso de uso, requerimiento de funcionalidad básica de la controladora.

Autoría propia.

Definido el caso de uso inicial se procedió a continuar con la segunda etapa de esta primer fase, la planificación, para esto nos apoyamos en el manejo de los software Project de

Microsoft y Trello de Atlassian, el primero nos permitió controlar el avance de los procesos relacionados a los diferentes eventos llevados a cabo a lo largo del desarrollo del proyecto, el uso de esta herramienta nos permitió mostrar con las fechas de ejecución cada uno de los eventos realizados a la gerencia de la compañía demostrando de esta manera el avance de cada uno de los procesos, ver anexos **B**.

En cuanto al manejo del software Trello este nos permitió agregar al coordinador del área y otras áreas afectadas de manera directa por los procesos que se desarrollaron a lo largo del proceso, esta herramienta nos permitió hacer seguimiento por medio de los dispositivos móviles de los procesos realizados en algunas etapas del proceso y validar directamente los avances realizados, guía en anexos **B**, esta herramienta además nos permitió integrar el uso de la metodología Kanban en los procesos de desarrollo y mantenimiento de los demás servicios ofrecidos no solo por el departamento de desarrollo.

En esta fase inicial los dos primeros procesos, la planificación y el diseño fueron los más extensos en relación con las 5 fases posteriores, esto dado por lo complejo que fue en un principio el poder recopilar las diferentes funciones desarrolladas por la controladora.

Posterior al avance de estas dos primeras se inició el proceso de codificación sin novedad alguna, las pruebas de esta primera fase se realizaron en las instalaciones de la compañía, contando con todas las herramientas necesarias, las novedades presentadas se fueron alimentando en el software Trello para que el ingeniero encargado de la codificación estuviera al tanto de los acontecimientos presentados.

En total el proyecto se dividió en 5 fases, de las cuales se obtuvieron tres programas funcionales, el primero de ellos se obtuvo al finalizar la fase 2 del proyecto, en este punto

del proyecto se entregó un firmware a grabar en la controladora con las funciones solicitadas por nuestro cliente interno, hasta la fecha este sistema se encuentra operando en la instalación designada por el departamento de proyectos de la compañía sin que se presenten problemas funcionales.

La segunda entrega del sistema operando se ha realizado al culminar la fase 4, en este punto el sistema ya cuenta con los requerimientos solicitados en la segunda historia de usuario, ver tabla 3.

Tabla 3

Segunda iteración del proyecto, mensualidades por boca.

Historia de Usuario	
Iteración: 2	
Cliente: Coordinadores de desarrollo y proyectos	
Nombre historia: Mensualidades por boca.	
Prioridad: Media	Riesgo: Alto
Programador responsable: Jaime Pedraza	
<p>Descripción: El transporte de tarjetas inteligentes debe capturar las tarjetas que se ingresen por la parte frontal, el transporte deberá poder identificar el tipo de tarjeta ingresada y validar cada uno de los puntos descritos a continuación:</p> <ol style="list-style-type: none"> 1. La tarjeta es tecnología MIFARE. 2. La tarjeta pertenece al estacionamiento. 3. La tarjeta es de rotación, de ser así retórnela y genere aviso informativo. 4. La tarjeta es mensual, deberá validar la información grabada en ella y si está dentro de los parámetros de autorización para el ingreso cambiará el estado de la tarjeta dejándola activa para la salida, debe dar apertura. Si la tarjeta incumple alguno de los requisitos retornarla y mostrar mensaje informativo. <p>Adicional a la validación realizada con las tarjetas ingresadas por la boca, debe mantener la operatividad con los sistemas AccesScan y monitor.</p>	

Observaciones: La funcionalidad para mensualidades de aproximación debe permanecer intacta.

Se realizará una segunda entrega funcional validando solo las funciones acá detalladas.

La tabla 3 permite conocer las funcionalidades solicitadas para el manejo de las mensualidades por inserción de tarjetas en el transporte. Autoría propia.

El caso de uso para esta iteración se puede apreciar en la figura 26.

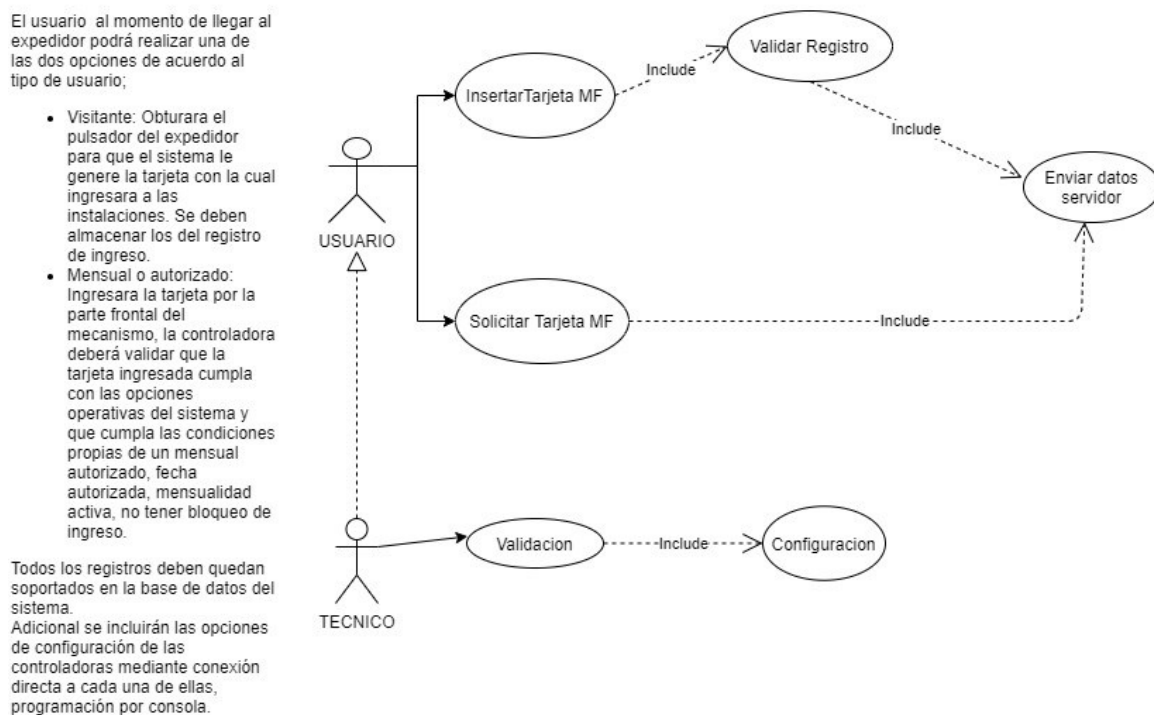


Figura 26. Segundo caso de uso, mensuales por boca. Autoría propia.

El sistema se encuentra operando y se continúan haciendo acciones de mantenimiento y puesta a punto para que se acople a las funciones del sistema actual de la instalación.

La última iteración realizada con las directivas de la compañía y con los coordinadores de los departamentos de desarrollo y proyectos es evidenciada en la tabla 4.

Tabla 4.
Tercera iteración del proyecto, validación de horarios.

Historia de Usuario	
Iteración: 3	
Cliente: Coordinadores de desarrollo y proyectos, gerencia general	
Nombre historia: Mensuales por boca y con horarios.	
Prioridad: Alta	Riesgo: Alto
Programador responsable: Jaime Pedraza	
<p>Descripción: Dentro de las funciones operativas de las controladoras se deberá poder integrar el manejo de horarios para los mensuales, estos horarios deberán estar acorde con los creados en el sistema AccesScan.</p> <p>Al ingresar la mensualidad en el transporte el programa deberá validar el horario que identifica el mensual y determinar si permite o no el ingreso conforme a los requerimientos del cliente, estos horarios están sujetos a cambios por lo tanto se podrán editar sin problema alguno.</p>	
Observaciones:	
<p>La tabla muestra el proceso final requerido por parte de los actores internos de la compañía, en esta última etapa se requiere integrar nuevas funcionalidades operativas en la configuración de la controladora. Autoría propia.</p> <p>Como resultado a esta última iteración se plantea el tercer caso de uso, en el cual se plasman las funciones finales a realizar por parte de la controladora, ver figura 27.</p>	

Se conservaran las funciones preestablecidas en el caso de uso anterior, se agrega la función de validación de los horarios de los mensuales, se deberá poder agregar los horarios en la configuración de la controladora para hacer su validación. Los horarios deben ser configurables ya que estos se modifican conforme lo establezca cada cliente.

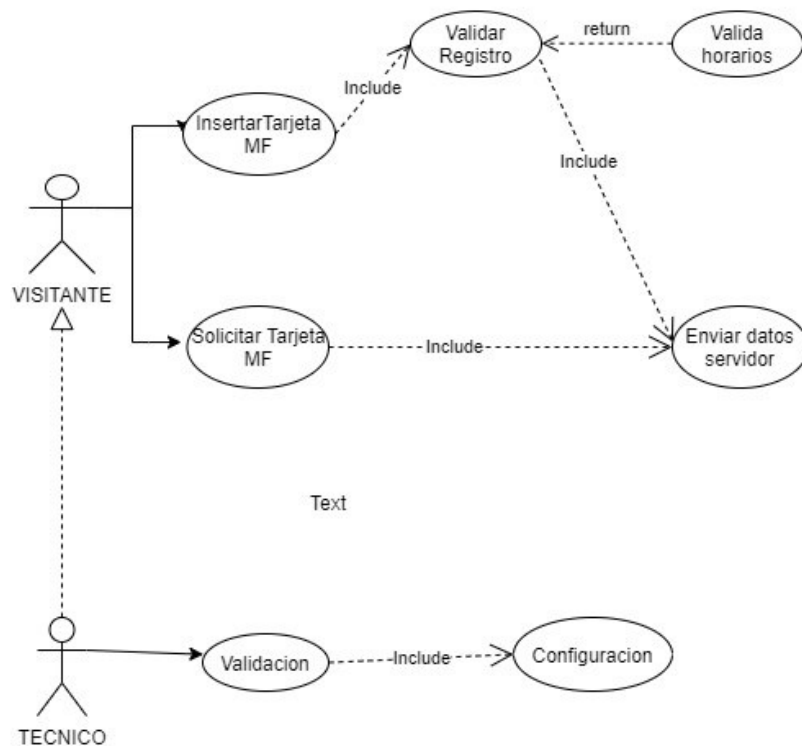


Figura 27. Tercer caso de uso, validación de horarios de mensuales. Autoría propia.

Mediante la aplicación de la metodología extrema XP y la integración de Kanban, se logró optimizar el tiempo de respuesta a las solicitudes realizadas por parte del cliente del sistema, en nuestro caso, el poder cumplir con las expectativas funcionales mínimas solicitadas por el coordinador del área de desarrollo y por la gerencia general de esta. Al implementar estos modelos de gestión de proyectos se pudo demostrar para las demás áreas de la compañía la importancia de contar con procesos definidos y estructurados a la hora de dar cumplimiento a cualquier tipo de solicitud.

B. Herramientas de seguimiento

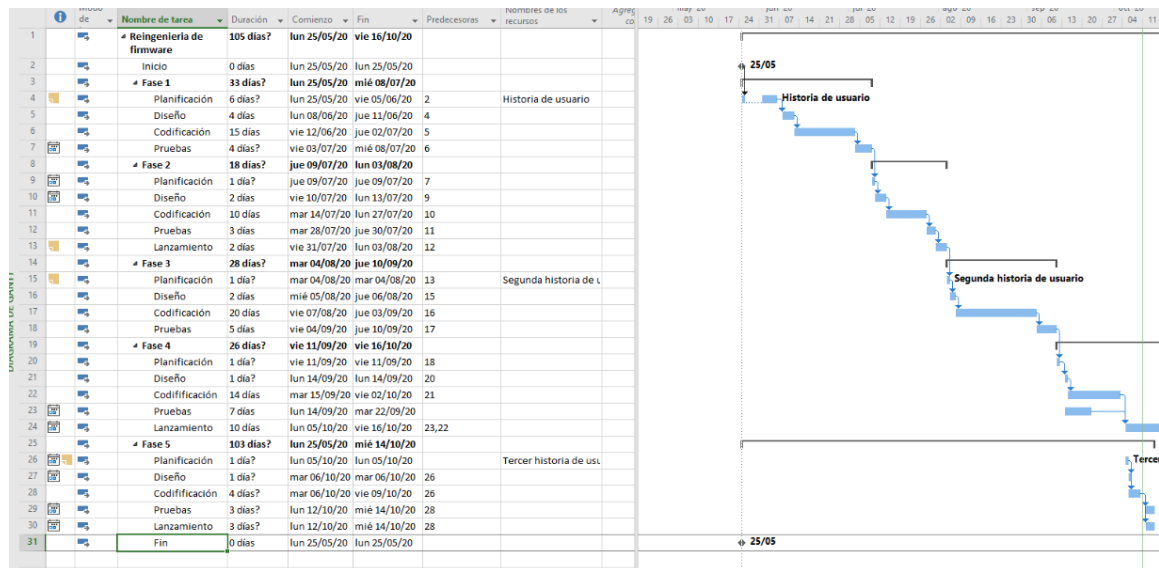


Figura 28. Captura de proceso implementado en software Project. Diseño propio.

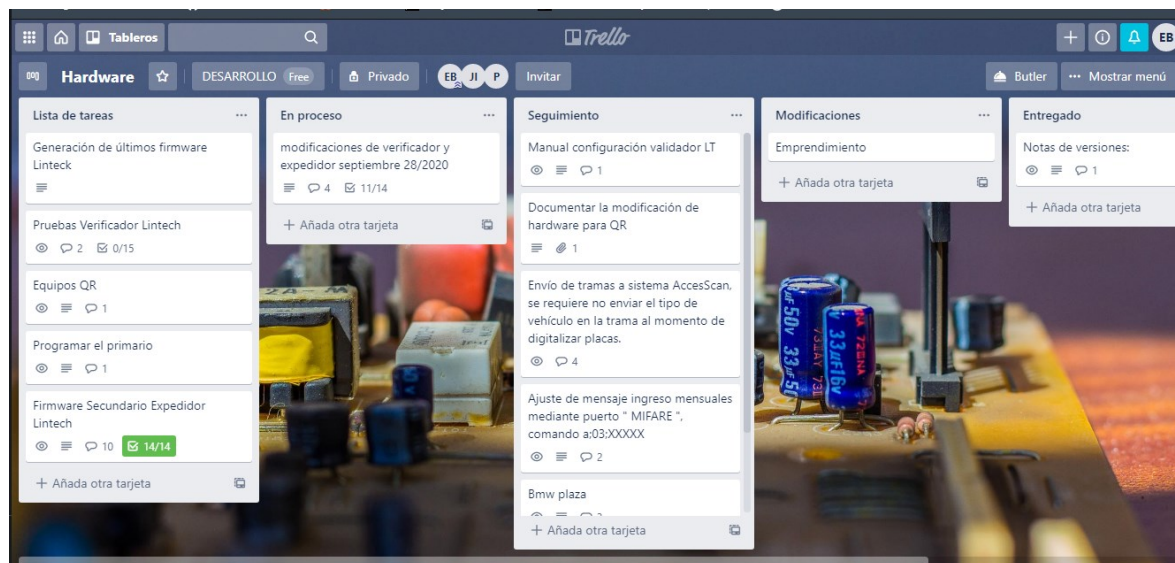


Figura 29. Tablero de software Trello implementado durante el proyecto, Adaptado de (Hardware | Trello, s. f.)

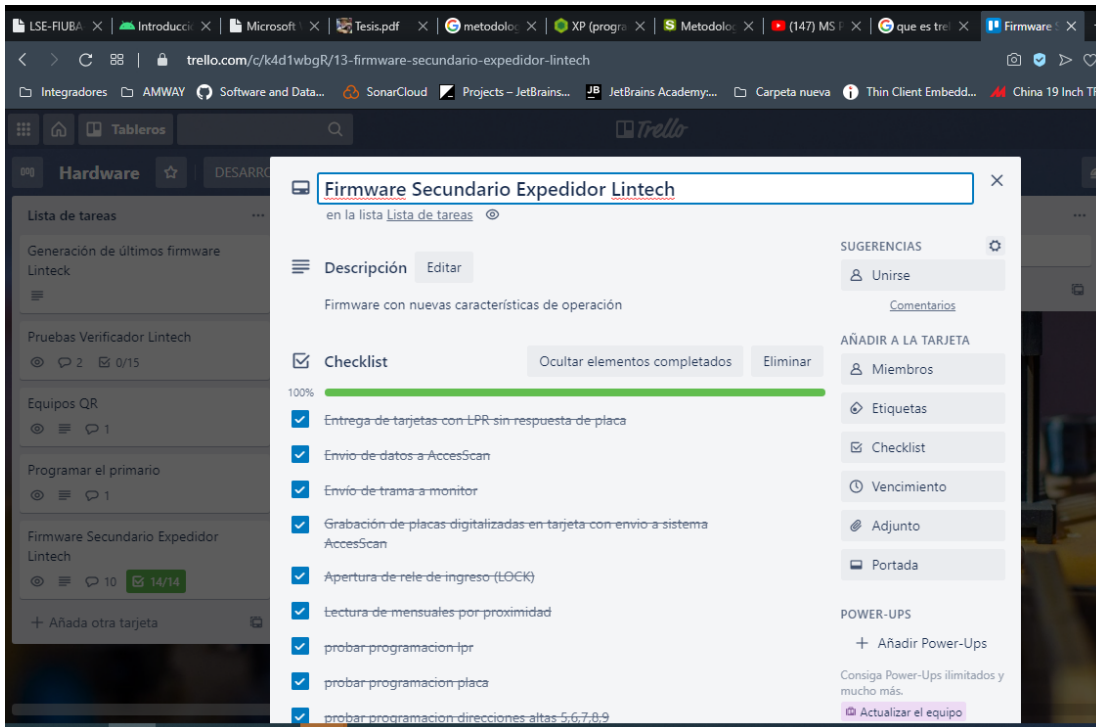


Figura 30. Lista de chequeo implementado en Trello para el seguimiento de las operaciones a realizar por el equipo. Adaptado de (Hardware | Trello, s. f.).

C. Protocolo de encriptación de claves

Para generar este codificador se creó una función que recibe un String de 10 caracteres que será codificado por un método propio que posee 3 tablas de 16 caracteres de la tabla ASCII que pueden ser del 0 al 256. Una de las tablas genera un aleatorio por cada miembro de la cadena a codificar, la segunda tabla con el aleatorio procesa una operación matemática y genera un nuevo aleatorio que es pasado por la tercera tabla que procesa una operación matemática diferente generando un carácter que a su vez es pasado por un nivel de comparaciones y así se genera el primer carácter de la clave que puede ser Alpha-numérico mayúsculas o minúsculas y así sucesivamente para los demás.

Para calcular la entropía “incertidumbre” de nuestro algoritmo se usa la siguiente formula

$$\log_2(1.208.925.819.614.629.174.706.176) = 80$$

El resultado de este cálculo es $= 2^{80}$ posibilidades de la clave para ser vulnerado el criptograma.

Se pasa por la calculadora de Shannon para probabilidades de eventos, figura 31.



Figura 31. Calculadora de eventos de Shannon, probabilidades de acceso al módulo de configuración mediante el ingreso de datos al sistema. Tomado de (Calculadora en línea: Entropía de Shannon, s. f.).

La segunda calcula la frecuencia de símbolos dada para un mensaje dado, figura 32.

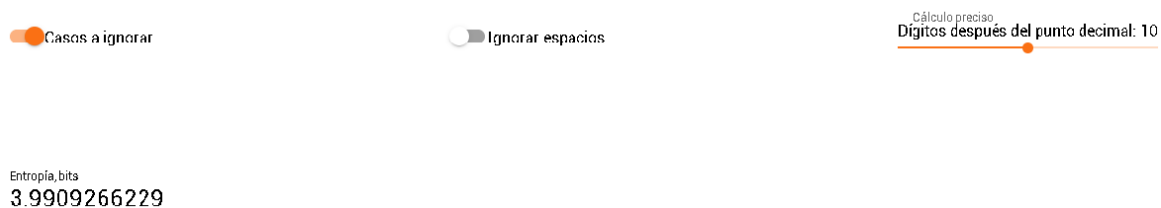


Figura 32. Calculadora de Shannon. Tomado de (Calculadora en línea: Entropía de Shannon, s. f.)

D. Resultados obtenidos

La implementación del nuevo firmware durante los últimos dos meses de validación y seguimiento realizado nos ha permitido identificar beneficios en diferentes aspectos, los más relevantes han sido:

- Disminución de los servicios técnicos.

La compañía requería con suma urgencia realizar el cambio de tecnología en los transportes de tarjetas inteligentes no solo por la falta de equipos para su reemplazo, también las constantes fallas operativas presentadas en varios de los puntos era una de

las necesidades primordiales, pero debido a la concepción del anterior sistema encontrar fallas de operación relacionados con el estado de los mecanismos o el estado de las tarjetas electrónicas podría llegar a tomar bastante tiempo, más aun si el personal técnico no contaba con las herramientas o las ayudas para realizarlo en campo. Una de las fallas más reportadas consistía en la entrega de más de una tarjeta a los visitantes, ocasionando demora en los procesos de ingreso y pérdida de las tarjetas, lo cual se reflejaba en costos adicionales a la compañía ya que, en la mayoría de los casos, estas tarjetas debían ser reemplazadas. Una instalación de mediana rotación vehicular podría reportar en un día una pérdida aproximada de cinco tarjetas MIFARE, con las actuales modificaciones no se han recibido notificaciones de este tipo en las dos instalaciones en las cuales se encuentra instalado el nuevo firmware. El detalle de este seguimiento se representa en la figura 33.

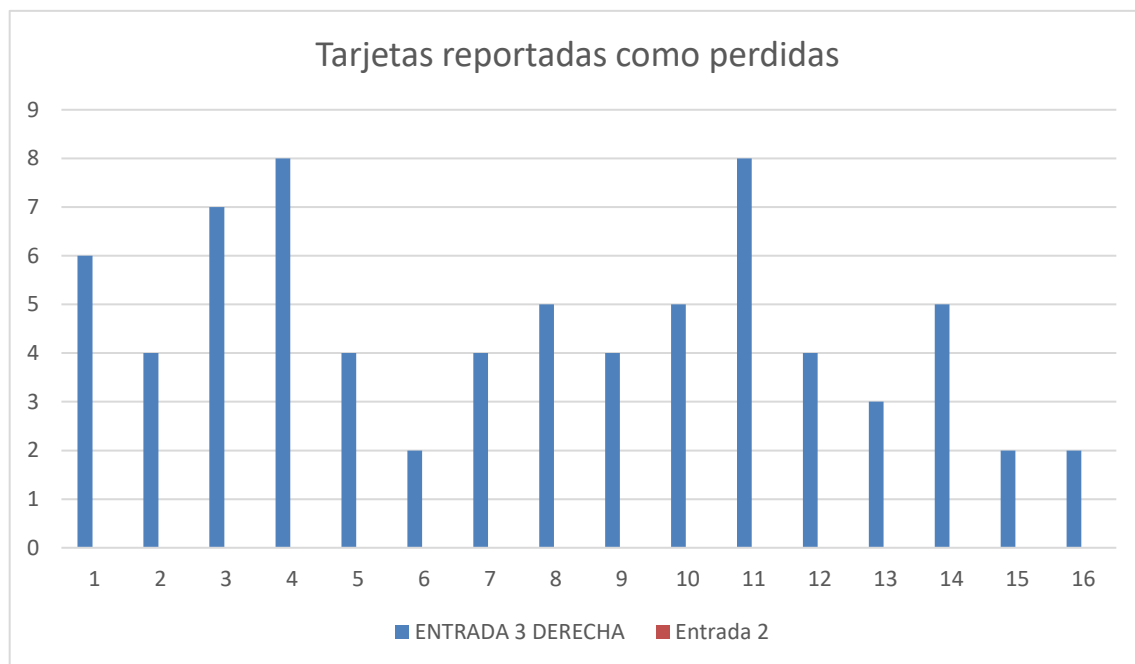


Figura 33. Detalle de tarjetas reportadas como perdidas. Autoría propia.

Otro de los problemas reportados con frecuencia era el relacionado al atasco de tarjetas de usuarios mensuales, en la figura 34 se detalla el comportamiento reportado por parte del personal operativo de la instalación.

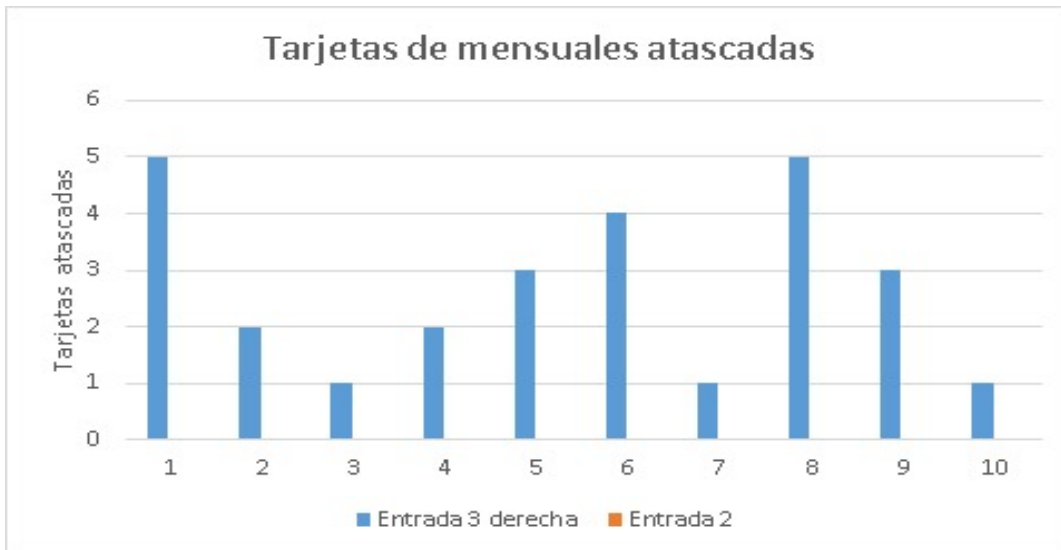


Figura 37. Reporte de tarjetas atascadas en los transportes, la entrada 2 corresponde a un ingreso actualizado, mientras que la entrada 3 hace parte de un equipo antiguo. Autoría propia.

- Rapidez en adaptarnos a las necesidades del cliente.

Con el anterior firmware se requería hacer el levantamiento de las configuraciones a realizar en la instalación, se codificaba y este firmware era enviado al técnico para su grabación, pero si las configuraciones no eran las adecuadas se requería repetir el proceso, causando pérdidas de tiempo y baja adaptabilidad a los requerimientos de los clientes. Con la implementación del módulo de configuración el técnico de soporte puede realizar el proceso directamente en los equipos, agilizando los tiempos de respuesta ante el cliente y dando la posibilidad de realizar ajustes sin requerir la intervención del área de desarrollo de la compañía.

E. Análisis de gestión de negocio

El análisis de la gestión del negocio se hace comparando el firmware antiguo en los meses de Julio y agosto y el nuevo firmware Septiembre y octubre.

El número de servicios presentado en el firmware antiguo son 20 servicios y presenta pérdida de las tarjetas MF50 de 5 unidades por servicio evidenciando que en el nuevo no hay pérdidas de tarjetas MF50, además los equipos son parametrizados con los nuevos métodos creados de configuración que se adaptan a la necesidad del cliente, reduciendo los costos de desarrollo a (0) en los nuevos firmwares. Con todos estos cambios el número de servicios es reducido a un 98%. en la tabla 5, se representan los costos en dólares de la operación de mantenimiento.

Tabla 5

Costos operativos, comparación de gastos antes y después.

Estado de resultados	Julio agosto 2020		Septiembre octubre 2020	
		Costos		Costos
Número de tarjetas perdidas por servicio	5		0	
numero de mantenimiento	20		2	
Costo de mantenimiento Ingresos	400	20	40	20
Costos Técnicos	42,4	2,12	4,24	2,12
Costo Materia prima	442,4	22,12	44,24	
Gastos administrativos	15,2	0,76	1,52	0,76
Gastos de Coordinación mantenimiento	20	1	2	1
Gastos de Desarrolló	40	2	4	2
Costo reposición tarjetas MF50	150	1,5	0	1,5
Costo Operaciones	667,6	33,38	51,76	25,88

La tabla nos permite apreciar el detalle económico de las actividades realizadas antes y después de realizado el proceso de actualización del firmware en los equipos de una de las instalaciones de la compañía. Autoría propia.

En la tabla 6 sacamos los datos de comparación de los servicios de firmware nuevo y el antiguo.

Tabla 6
Costos operativos

	costo fijo total	costo fijo promedio	costo variable	costo variable	costo total	costo total promedio	cambio en el costo	costo marginal	producción marginal
Numero Mantenimientos	CFT	CFP	CVT	CVP	CT	CTP	ACT	CMg	
2	44,24	22,12	7,52	3,76	51,76	25,88	615,84	307,92	2
20	442,4	22,12	225,2	11,26	667,6	33,38	51,76	2,875	18

La tabla nos da a conocer el valor económico implicado en cada proceso de mantenimiento llevado a cabo por la compañía durante la atención de fallas reportadas, información suministrada por el departamento contable de la compañía. Autoría propia.

En la figura 35 tenemos el número de mantenimientos, podemos apreciar en costo total (CT) teniendo unos costos con el firmware antiguo de \$667.6 USD y en los nuevos el cambio en el costo total (ACT) un ahorro de \$615,84 USD y viendo una reducción en los servicios (CMg) de 18 servicios con el firmware nuevo. La pérdida de tarjetas inteligentes en el nuevo firmware es de (0). La reducción del costo (CTP) o el costo promedio del servicio con el firmware nuevo es de \$25,88 USD para el nuevo y el antiguo un costo de \$33,38 USD. Con esto se puede evidenciar las mejoras implementadas

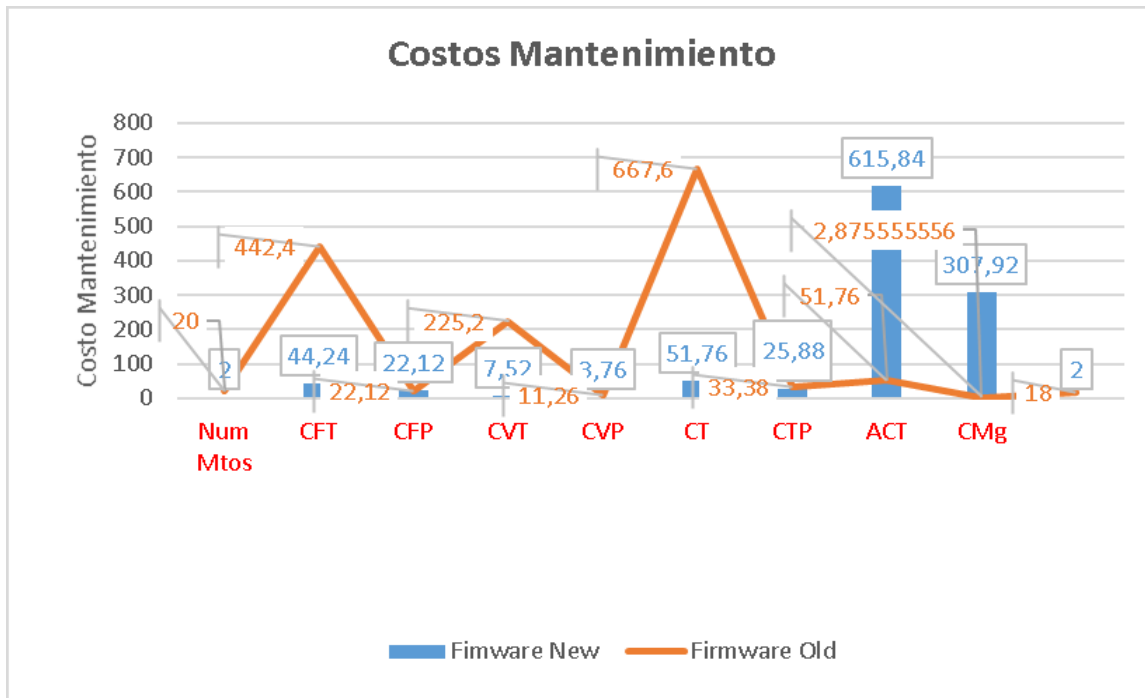


Figura 41. Gráfica demostrativa de la reducción de costos obtenidos a partir del proceso de actualización realizado, información suministrada por el departamento contable de la compañía. Autoría propia.

8. Conclusiones

Cumpliendo a cabalidad con todo el proceso de reingeniería del firmware de la tarjeta

AccesScan se tiene las siguientes recomendaciones y conclusiones:

- Se logró identificar las necesidades fundamentales que existían en la parametrización de configuración en los diferentes parqueaderos, personalizando cada configuración a la necesidad del cliente.
- Partiendo de un proceso de investigación interna, se logró identificar los ítems a mejorar, tomando como referencia el estado del código heredado y definiendo el estado operativo del nuevo firmware a implementar.

- Se diseñó de manera eficiente la estructura del software empotrado en un microcontrolador, la cual permite tener orden y reglas de calidad de software y Seguridad seleccionando las herramientas más adecuadas, según las necesidades y expectativas
- Se aplicó un software para pruebas unitarias empotrado en el microcontrolador que cumplió con todas las expectativas de prueba y se pudo aislar y localizar los errores. alcanzando los objetivos planteados en el proceso de inspección y control.
- Se recomienda que la organización siga utilizando e implementando todas las técnicas de arquitectura contempladas, para facilitar la labor de planeación estratégica. He incentivar las buenas prácticas de desarrollo en calidad y seguridad.
- Se recomienda que una vez el aplicativo llegue a una madures se haga un refactoring permitiendo disminuir esfuerzos como también ahorrar tiempo en los procesos de desarrollo y mejorar la mantenibilidad.
- Se recomienda que la organización siga implementada PMI en los proyectos para fomentar las mejores prácticas de gestión teniendo en cuenta que ayuda a la organización a tener mejores resultados en tiempos, costos, calidad, alcance, recursos humanos, comunicaciones, integridad y conocer los interesados así podremos tener una comunicación más efectiva y llegar a los verdaderos objetivos del proyecto.

9. Referencias – Bibliografía

- Blanes, R. G. (s. f.). El Libro Práctico del Programador Ágil: Un enfoque integral y práctico para el desarrollo de software mediante las mejores prácticas de código limpio, ... de diseño y gestión de la configuración. Rafa G. Blanes.
- Calculadora en línea: Entropía de Shannon. (s. f.). Recuperado 5 de noviembre de 2020, de <https://es.planetcalc.com/2476/>
- Canive, T., & Balet, R. (s. f.). Metodología XP o Programación Extrema: ¿Qué es y cómo aplicarla? Gestor de proyectos online. Recuperado 26 de octubre de 2020, de <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp>
- Cardacci, D. (2016). REFACTORIZACIÓN DE CÓDIGO Y CONSIDERACIONES SOBRE LA COMPLEJIDAD CICLOMÁTICA. UCEMA.
<https://ucema.edu.ar/publicaciones/download/documentos/592.pdf>
- Conesa, M. J. G., & Contreras, M. F. R. (2013). Desarrollo Software para la Gestión de un Terminal Punto de Venta. 79.
- Cuadros, D. (2015). Commits—Administrar tu repositorio. [Educación]. CódigoFacilito.
<https://codigofacilito.com/articulos/commits-administrar-tu-repositorio>
- Dispositivos lógicos programables—EcuRed. (s. f.). Recuperado 29 de noviembre de 2020, de https://www.ecured.cu/Dispositivos_l%C3%B3gicos_programables
- Hardware | Trello. (s. f.). Recuperado 5 de noviembre de 2020, de <https://trello.com/b/OjkCQnyT/hardware>
- Murillo, A. (2013). CTIN. <http://www.ctinmx.com/que-es-un-plc/>