SEGURIDAD MEDIANTE RFID UTILIZANDO ALGORITMOS DE ASIGNACION DE ACCESS KEY

CAMILO ALBERTO CEPEDA MEJÍA

DIEGO FERNANDO RODRÍGUEZ CASTAÑEDA

CRISTIAN CAMILO TRIANA REDONDO

TRABAJO DE GRADO PRESENTADO PARA OPTAR POR EL TITULO DE TECNÓLOGO EN DESARROLLO INFORMATICO

UNIVERSIDAD ESCUELA COLOMBIANA DE CARRERAS INDUSTRIALES

FACULTAD DE INGIENERIA

BOGOTÁ, D.C.

2017

Contenido

Lista de ilustraciones

Lista de tablas

SEGURIDAD MEDIANTE RFID UTILIZANDO ALGORITMOS DE ASIGNACION DE ACCESS KEY

Identificación del problema

Descripción del problema

La seguridad informática es uno de los requisitos claves dentro de cualquier institución u organización.

Actualmente la Universidad ECCI no tiene la protección adecuada en la lectura de los dispositivos biométricos, conclusión obtenida posterior a un estudio realizado en el semillero de internet de las cosas (IoT), utilizando electrónica a bajo costo. En este estudio se evidenció que los carnés utilizados por los estudiantes para el acceso e identificación dentro de las instalaciones, no cuentan con seguridad en su lectura ni en su escritura, por lo que se facilita el acceso a su información y la reproducción de copias de la misma, ocasionando un problema serio de seguridad en el acceso a las instalaciones de la Institución.

La realización de pruebas de lectura a los carnés, permitió conocer que poseen la codificación por defecto de los dispositivos RFID (Radio Frequency Identification), generando un fácil acceso a su información por parte de personas con conocimientos en el tema de informática.

La necesidad de solucionar estos problemas de seguridad, se ha convertido en el tema principal a desarrollar con esta investigación, mediante un sistema óptimo que implica mejoras en velocidad de acceso y optimización del algoritmo de seguridad utilizado

actualmente. La propuesta incluye un software con un sistema de selección de claves de acceso más aleatorio, para lograr evitar repeticiones en las tareas de los carnés.

La población a la que se encuentra orientada el proyecto está dividida en dos sectores: el corporativo y los usuarios finales. El primero corresponde a las áreas de registro y control y de seguridad, encargadas de la asignación del ID y la activación del carné. El segundo, hace referencia a la comunidad académica quienes portan y utilizan el carné.

En el aspecto económico, el desarrollo de esta propuesta en el semillero de investigación, se convierte en un beneficio para la Universidad, como un producto que evidencia la apropiación y posterior transferencia de conocimiento a la comunidad universitaria.

Formulación del problema

¿Cómo optimizar la seguridad del sistema de acceso con carné a la Universidad ECCI?

Objetivos

Objetivo General

Diseñar un módulo de control de acceso basado en un algoritmo de asignación de claves, el cual permitirá un mejor nivel de seguridad en dispositivos de accesos biométricos en la Universidad ECCI.

Objetivos Específicos

1- Planear y estudiar nuevos algoritmos y tecnologías aplicables.
2- Ejecutar pruebas en dispositivos y algoritmos actualmente implementados para evidenciar posibles fallos y deficiencias.
3- Realizar un algoritmo de asignación de claves según la hora del día.
4- Diseñar una directiva de seguridad que optimice la que actualmente utiliza la Universidad.
5- Proponer soluciones ante los errores presentados por el actual sistema.

Justificación y delimitación

El siguiente proyecto está orientado a mejorar la seguridad informática en accesos biométricos de la Universidad ECCI, mediante un algoritmo basado en el cambio de claves de acceso según la hora, por medio de los dispositivos de comunicación RFID que utiliza actualmente la institución.

Este algoritmo de cambio de clave está proyectado para dispositivos biométricos tales como torniquetes, que al momento se encuentra probado en el módulo de lectura RFID, MFRC522[1], soportado por Raspberry, el cual emula los registros tomados por un torniquete normal. Con este algoritmo de cambio de clave se realizan las tareas básicas de lectura y escritura de las tarjetas para emular el ingreso y egreso del personal. Lo anterior, hace parte de un avance semi funcional del proyecto buscando su implementación a largo plazo en la Universidad, por lo que es necesario realizar pruebas con los estudiantes de la Universidad, garantizando así mejoras en la seguridad de los dispositivos de RFID.

Además del cambio en las claves de acceso mencionado anteriormente, se realizó un análisis de los diferentes estándares propuestos para el manejo de seguridad de dispositivos RFID. Entre éstos podemos encontrar los siguientes: ISO 10536 que especifica los materiales y condiciones que deben tener las tarjetas, por ejemplo, la rigidez a la flexión, toxicidad, resistencia al calor, entre otros, ISO 15693 que maneja un frecuencia igual a las tarjetas RFID siendo de 13.56MHz pero a una mayor distancia, e ISO 7816 que especifica las características físicas de las tarjetas tales como dimensiones y sus toleración, y condiciones para la conformidad, los tres basados en proximidad y lectura de dispositivos RFID enfocado a tarjetas o para fines del proyecto carnés. Con este estudio se llegó a la conclusión de la utilización del actual estándar de la universidad (ISO 14443) porque es el estándar ISO que actualmente está en uso, realizando un cambio al módulo de seguridad implementado en la Universidad.

[1] Lector MFRC522: https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf

El proyecto pretende sugerir cambios significativos a futuro, abarcando el ámbito de la seguridad desde el punto de vista lógico, realizando pruebas de lectura que mejoren la eficacia en los procesos actuales y optimicen las tareas de acceso, estableciendo así parámetros convenientes para la Universidad.

Lo anterior, basado en pruebas que demuestran la optimización del sistema actual, logrando así un proceso de lectura de tarjetas más riguroso.

Marco referencial

Marco teórico

A continuación, se dará lugar al marco teórico, el cual busca identificar y explicar los conceptos claves del proyecto, haciendo énfasis en la función de estos términos, además de su importancia dentro del desarrollo de la temática RFID, que es el tema central del documento.

Se expondrán temas como, sistemas de información, RFID, Raspberry Pi, Comercio y tecnologías móviles, IoT, Python etc. Todos los conceptos mencionados tienen una aplicación dentro del proyecto.

Sistema de Información

Los sistemas de información son un compendio de elementos que se utilizan para manipulación, ordenamiento y administración de información solucionando una necesidad. Contienen bases de información de varios temas y con características diversas, sus datos manejan una relación que busca ordenarlos y clasificarlos en conjunto. Éstos se diseñan para tener gran almacenamiento y permitir mejoras en la administración del contenido, por consiguiente, son utilizados en las empresas con el propósito de mejorar la gestión en sus

inventarios y evitar errores de conteo, o simplemente para reducir la probabilidad que un fraude se pueda producir.

Sabiendo que la tecnología actual es usada para facilitar el trabajo de control y gestión gracias a que procesa una gran cantidad de información en poco tiempo y usando pocos recursos, se recoge la necesidad de crear un sistema de información para mejorar la competitividad en cualquier sector, por ejemplo, la necesidad de un hotel para almacenar su información de manera segura y logrando que esta esté fácilmente accesible.

Dentro de cualquier empresa, el manejo de la información es la clave en la organización como dice Effy Oz "en los negocios, las personas y las organizaciones buscan utilizar la información de manera específica para tomar decisiones sólidas y para resolver problemas" (Oz, 2008). La organización es una de las claves para el manejo de un sistema, por lo que su cuidado es una de las prioridades en la creación de las bases de datos, si esta se mantiene en un constante ordenamiento y actualización permitirá cualquier consulta que se realice en el futuro, además se debe asegurar que el almacenaje de esta información no afecte a ningún usuario, que pueda ser filtrada y ser usada con fines maliciosos, por eso debe estar bajo las leyes actuales que rigen el uso de la información. Se necesita aplicar todos estos elementos al sistema que se creará en una empresa o proyecto ya que otorgará seguridad a la información que pueda ser adquirida por parte de los usuarios.

Raspberry Pi

Raspberry Pi es un computador a bajo costo que tiene como cualidad única una gran capacidad para su reducido tamaño y costo, pero como todo computador este no sería más que un simple circuito sin el complemento de un software que realice todas las operaciones que cumple un sistema operativo. Dentro del cual se generarán todas y cada una de las tareas que se planean ejecutar dentro del sistema de seguridad informática.

Este computador es el módulo electrónico de preferencia en lo que respecta al contexto de automatización en dispositivos a bajo costo, gracias a que sus puertos GPIO permiten interactuar de manera directa con los módulos electrónicos que se requieran para el sistema.

Dennis argumenta lo siguiente:

Además, gracias a la misión de Raspberry Pi de proporcionar una herramienta educativa para aquellos interesados en la programación, la adición de Arduino proporcionará un mecanismo para aquellos que deseen pasar de software de escritura que manipula la Raspberry Pi, a un software que manipula su entorno. Y proporciona una vía para aprender sobre la electrónica. Esto podría tener el efecto positivo de reforzar las filas de los clubes home-brew y Maker con un ojo hacia la automatización del hogar y llevar a una diversidad cada vez mayor de herramientas que se producen para el público. (Dennis, 2013).

En un entorno educativo, Raspberry Pi puede ser utilizado gracias a su fácil manejo, el cual involucra diversas áreas de la ingeniería como la programación y la electrónica.

Gracias a sus facilidades como herramienta académica Raspberry posee una gran cantidad de proyectos, automatización, seguridad, herramientas biométricas, circuitos complejos etc.

Sistema Operativo

Un sistema operativo es un software plataforma que permite el control del hardware de un dispositivo, y las distintas operaciones que este tiene, en el panorama tecnológico actual hay una gran variedad de softwares con estas funciones, varios ejemplos pueden ser Windows como referente más famoso de esto debido a su fácil manejo para todo tipo de usuarios. Linux es otro de los sistemas operativos importantes dentro de esta industria, este tiene varias características que lo hacen muy propicio para funciones más avanzadas como programación, hacking o manejo de redes, pero lo que hace a Linux especial es su filosofía de código abierto Open Source que se basa en el concepto de un software gratuito con todo tipo de posibilidades para su modificación.

Open Source fue una revolución en la industria del desarrollo de software, debido a que los usuarios de un computador tenían un repertorio más amplio para escoger nuevos modelos de sistema operativo, que cumplieran las mismas funciones que realizaban los sistemas monopolizados y lo mejor adquirirlos de manera gratuita.

Open Source posee varias características que lo hacen una gran opción a la hora de escoger sus productos de software, por su estado de gratuidad y la posibilidad de manejar a placer el código fuente de cualquier software (Esteve, 2014):

a) Acceso al código fuente, ya sea para estudiarlo (ideal para educación) o modificarlo, sea para corregir errores, adaptarlo o añadir más prestaciones.
b) Gratuidad: normalmente, el software, ya sea en forma binaria o en la forma de código fuente, puede obtenerse libremente o por una módica cantidad en concepto de gastos de empaquetamiento, distribución y valores añadidos.
c) Evitar monopolios de software propietario: no depender de una única opción o único fabricante de nuestro software. Esto es más importante cuando se trata de una gran organización, ya sea una empresa o estado, los cuales no pueden (o no deberían) ponerse en manos de una determinada única solución y pasar a depender exclusivamente de ella.
d) Un modelo de avance, no basado en la ocultación de información, sino en la compartición del conocimiento (semejante al de la comunidad científica), para lograr progresos de forma más rápida, con mejor calidad, ya que las elecciones tomadas están basadas en el consenso de la comunidad, y no en los caprichos de empresas desarrolladoras de software propietario.

RFID

RFID (Radio Frequency Identification) está basado en la utilización de dispositivos biométricos los cuales se comunican mediante la trasmisión de radio frecuencias generando una codificación en formato hexadecimal que se distribuye en patrones utilizados como keys de entrada, los cuales, al ser reconocidos dentro de una base de datos, generaran accesos cumpliendo con las características que requiere un sistema de seguridad físico y lógico.

Todo el concepto de lectura y trasmisión de datos por radio frecuencia no es un concepto reciente, lleva varios años en el campo tecnológico, siendo referenciado como una herramienta de alta utilidad. Un claro ejemplo al respecto, es la utilización de este protocolo

en el sistema Transmilenio, el cual recibe, analiza y guarda información por medio de tarjetas y torniquetes.

Adicionalmente, la información que guardan los sistemas biométricos también ayuda a crear un sistema con varias categorías. En el caso de Transmilenio se puede evidenciar un cambio en el costo del servicio según ciertos parámetros, como el estrato social o la edad del usuario.

RFID es un estilo tecnológico que está tomando un lugar importante en la industria, su capacidad de lectura lo hace óptimo para tareas como registro, seguridad etc. Como ejemplo se tiene que RFID cuenta con un sistema dentro del mercado de dispositivos móviles (Smartphone), el cual tiene como cualidad la lectura y retorno de datos lo cual permite a los usuarios explorar gran cantidad de productos desde sus celulares o dispositivos móviles, generando un gran cambio en el mercado de todo tipo de productos.

Comercio y tecnologías móviles

El comercio-m (comercio móvil) es un área en la que los teléfonos móviles están comenzando a utilizarse (Senn, 2000). Los mensajes cortos de texto del dispositivo móvil se utilizan para autorizar pagos de alimentos en las máquinas expendedoras, boletos del cine y otros artículos pequeños en vez de usar efectivo y tarjetas de crédito. Posteriormente el cargo aparece en la factura del teléfono celular. Cuando el dispositivo móvil está equipado con tecnología NFC (Comunicación de Campo Cercano, del inglés Near Field Comunication), puede actuar como una tarjeta inteligente RFID e interactuar con un lector cercano para realizar un pago. (Singh, 2016).

Singh habla de la comercialización del mercado móvil y su incursión dentro de las tecnologías RFID, nuestros celulares son un claro ejemplo de lo anterior mencionado, lectores NFC incorporados dentro de ellos los hacen más versátiles, más cómodos en cuestión de pagos, transacciones o tecnologías de seguridad biométrica.

IoT

IoT es una tendencia que se basa en el concepto de interconexión de las cosas u objetos mediante internet, IoT tiene como funcionalidad la integración de tecnologías Open Source, diseños en código abierto y su unión con hardware a bajo costo, esto está descrito por Salazar y Castro "En este quinto artículo de nuestra serie IoT, observamos los muchos nuevos sistemas operativos de código abierto que apuntan al IoT. Nuestras publicaciones anteriores han examinado los marcos de código abierto IoT, así como Linux y hardware de desarrollo de código abierto para IoT y dispositivos de consumo doméstico inteligente. Pero todo comienza con el sistema operativo.

En la última década, la mayoría de los nuevos proyectos de software de código abierto han pasado del mercado móvil a Internet de las cosas. En este quinto artículo de nuestra serie IoT, observamos los muchos nuevos sistemas operativos de código abierto que apuntan al IoT. Nuestras publicaciones anteriores han examinado los marcos de código abierto IoT, así como Linux y hardware de desarrollo de código abierto para IoT y dispositivos de consumo doméstico inteligente. Pero todo comienza con el sistema operativo.

Además de explorar nuevas distribuciones basadas en Linux incorporadas basadas en IoT, he incluido algunas distribuciones ligeras más antiguas como OpenWrt que han visto renovada absorción en el segmento. Aunque las distribuciones de Linux están dirigidas principalmente a los gateways y hubs, ha habido un crecimiento equivalente en sistemas operativos de código abierto no Linux, que pueden funcionar en unidades de microcontroladores (MCU), y están dirigidos típicamente a dispositivos de borde IoT.

Tenga en cuenta que casi todos los sistemas operativos de este día están reclamando alguna conexión IoT, por lo que la lista es algo arbitraria. Los candidatos aquí cumplen la mayoría de las siguientes propiedades: huella de memoria baja, alta eficiencia de energía, una pila de comunicación modular y configurable, y un fuerte soporte para tecnologías inalámbricas y sensores específicos. Algunos proyectos hacen hincapié en la seguridad IoT, y muchos de los sistemas operativos Linux se centran en el determinismo en tiempo real,

que a veces es un requisito en IoT industrial. (Estados Unidos Patente nº US5802467 A, 1998)

Para el proyecto la finalidad de involucrar IoT es buscar el enfoque dentro del campo de la seguridad informática más específicamente la seguridad en los dispositivos biométricos de la Universidad ECCI con referencia a Salazar y Castro (1998) que argumentan "Algunos proyectos hacen hincapié en la seguridad IoT, y muchos de los sistemas operativos Linux se centran en el determinismo en tiempo real, que a veces es un requisito en IoT industria".

En este proyecto se expone el hecho de que IoT llegará a tener un gran auge en un futuro, esta tecnología está en búsqueda de una tarea, implementar internet a todos los objetos posibles, haciendo de ellos dispositivos más funcionales y más versátiles que brinden una sensación de comodidad en todos los entornos donde la gente los utilice, este es el concepto básico de Internet de las cosas por sus siglas IoT.

Python

Python es un lenguaje de programación que surgió de la necesidad de trabajar bajo el concepto de multiplataforma siendo este una plataforma en términos más simples un sistema operativo, también tiene como característica su función como multiparadigma, que le permite realizar programación de distintas formas, entre ellas, programación orientada a objetos, a mediano y bajo nivel, estructurada y no estructurada etc. Esta versatilidad esta explicada por Jones en su libro Python for Unix and Linux System Administration "Hicimos un esfuerzo concertado para crear ejemplos que realmente le ayudarán a hacer su trabajo. Hay ejemplos de maneras de descubrir y supervisar subredes automáticamente con SNMP, para convertir a una shell interactiva de Python llamada IPython, para construir tuberías de procesamiento de datos, para escribir herramientas de gestión de metadatos personalizados con asignadores de objetos relacionales, para realizar la programación de red, para escribir comandos herramientas de línea, y mucho más.

Aunque Python se está encendiendo como un reguero de pólvora, hay muchos administradores de sistemas que han estado expuestos a Bash y Perl solamente. Si te encuentras en esta categoría, deberías tener consuelo en saber que Python es muy fácil de

aprender. ¡De hecho, aunque es una cuestión de opinión, Python es considerado por muchos como el idioma más fácil de aprender y enseñar, punto! (Jeremy Jones, 2009)

Python de más fácil aprehensión dentro de lo que respecta a escritura y codificación, según Jeremy Jones, por su actual auge, Python logro un incremento en los sectores académico, esto puede ayudar a que más personas reconozcan, aprendan y utilicen este lenguaje, lo cual fortalece su desarrollo y evolución.

Para este proyecto se utilizará Python, teniendo en cuenta las funciones de escritura y lectura de los dispositivos RFID. Aunque este lenguaje no es el único que brinda los servicios de conexión y edición que requieren los módulos relacionados con los puertos GPIO de la Raspberry, en cuestión de facilidad en la escritura y lógica del código y velocidad de tareas tiene mejores resultados, convirtiéndose en una buena opción dentro del panorama de programación.

Python relaciona las dos principales funciones lógicas de un dispositivo RFID, que comprenden la escritura y lectura en dichos dispositivos, ya que se pueden reescribir keys de entrada y editar sectores para que estos nunca más vuelvan a ser editables. Además, es posible lograr que todas las tarjetas lleven un registro claro y pueden tener los niveles de jerarquía.

Marco conceptual.

1- Python: Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

2- Raspberry: Es un computador de placa reducida, computador de placa única o computador de placa simple (SBC – Single Board Computer en inglés) de bajo coste desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza.

3- Módulos electrónicos: Es el sistema integrado que se utiliza para controlar las funciones del sistema eléctrico dentro de un sistema.

4- SGBD: Un sistema gestor de base de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de consulta y de generación de informes, o bien mediante aplicaciones al efecto.

5- GPIO: (General Purpose Input/Output, Entrada/Salida de Propósito General) Es un pin genérico en un chip, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede controlar (programar) por el usuario en tiempo de ejecución. Los pines GPIO no tienen ningún propósito especial definido, y no se utilizan de forma predeterminada. La idea es que a veces, para el diseño de un sistema completo que utiliza el chip podría ser útil contar con un puñado de líneas digitales de control adicionales, y tenerlas a disposición ahorra el tiempo de tener que organizar circuitos adicionales para proporcionarlos.

6- Domótica: Son los sistemas capaces de automatizar una vivienda o edificación de cualquier tipo, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar. Se podría definir como la integración de la tecnología en el diseño inteligente de un recinto cerrado.

7- IP (Internet Protocol): Es un número que identifica, de manera lógica y jerárquica, a una Interfaz en red (elemento de comunicación/conexión) de un dispositivo

(computadora, tableta, portátil, Smartphone) que utilice el protocolo IP (Internet Protocol), que corresponde al nivel de red del modelo TCP/IP

8- BD: (Base de datos) Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

9- Registro en una BD: En informática, o concretamente en el contexto de una base de datos relacionales, un registro (también llamado fila o tupla) representa un objeto único de datos implícitamente estructurados en una tabla. En términos simples, una tabla de una base de datos puede imaginarse formada de filas y columnas o campos. Cada fila de una tabla representa un conjunto de datos relacionados, y todas las filas de la misma tabla tienen la misma estructura.

10- Control de Cuentas: El Control de Cuentas de Usuario (UAC por sus siglas en inglés) es una tecnología e infraestructura de seguridad. Su objetivo es mejorar la seguridad al impedir que aplicaciones maliciosas hagan cambios no autorizados en el ordenador.

11- Multiplataforma: En informática multiplataforma es un concepto referido a programas informáticos o métodos y conceptos de cómputo que son implementados e interoperan en múltiples plataformas informáticas.

12- Algoritmo: Es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite llevar a cabo una actividad mediante pasos sucesivos que no generen dudas a quien deba hacer dicha actividad.

13- Driver: Un Driver, o controlador, es un programa que controla un dispositivo. Cada dispositivo, ya sea una impresora, un teclado, etc., debe tener un programa controlador.

14- Periféricos: Se consideran periféricos a las unidades o dispositivos de hardware a través de los cuales la computadora se comunica con el exterior, y también a los sistemas que almacenan o archivan la información, sirviendo de memoria auxiliar de la memoria principal.

Marco legal.

"La Ley Estatutaria 1266 del 31 de diciembre de 2008, por la cual se dictan disposiciones generales del habeas data y se regula el manejo de la información contenida en bases de datos personales, en especial la financiera, crediticia, comercial, de servicios y la proveniente de terceros países y se dictan otras disposiciones, era la tabla de salvación de cientos de miles de colombianos que vieron suprimida la posibilidad de acceder a un crédito en el sistema financiero formal, porque durante la crisis de los años noventa habían incurrido en mora o incumplido alguna de sus obligaciones crediticias, debiendo recurrir, en no pocas ocasiones, al sistema extra bancario con altas tasas de interés y poniendo en riesgo su mínimo patrimonio."

Habeas data, es la ley más reconocida por sus políticas de seguridad en la información digital de los colombianos. Esta ley cobija la información que se maneja en dispositivos RFID, teniendo en cuenta que en los carnés se utiliza información clasificada de estudiantes o administrativos. Por consiguiente, es importante controlar el acceso de terceros para evitar el uso indebido de esta información.

Gestión del proyecto de TIC

Fase de iniciación.

El objetivo principal es el de diseñar un sistema de seguridad que brinde una idea para una futura implementación del mismo, el límite del proyecto llego al punto de ser un

prototipo semi-funcional, dando como entrega el algoritmo de asignación de claves de acceso según la hora del día.

La población que se verá afectada dentro de la Universidad, teniendo en cuenta el nivel organizativo es el área de registro y control, realizando cambios en los sistemas biométricos registrados dentro de su base de datos (torniquetes, carnet).

Luego de un análisis del sistema actual podemos evidenciar los siguientes requerimientos:

Requerimientos funcionales del sistema

1- Realizar lecturas de tarjetas RFID.
2- Realizar escritura de tarjetas RFID.
3- Buscar la UID (Identificación Especifica) de las tarjetas.
4- Escoger sectores específicos de la arquitectura de los dispositivos según la hora del día.
5- Realizar conversiones de hexadecimal a binario y decimal.
6- Buscar las Access Key de las tarjetas mediante un algoritmo de fuerza bruta.
7- Otorgar formatos de fecha y hora.
8- Realizar búsquedas en bases de datos según el Access Key.
9- Otorgar acceso a las instalaciones de la Universidad.
10- Clasificar de manera jerárquica los carnets.

Requerimientos No Funcionales del Sistema

1- Mejorar el actual algoritmo de seguridad de la Universidad.
2- Generar lecturas de manera rápida y organizada de las tarjetas.
3- Mostrar error en caso de una lectura incompleta.
4- Funcionamiento 24/7.
5- Lectura de cualquier tipo de tarjeta RFID.
6- Operatividad multi-sistema.
7- Compilación y código en Python.
8- Uso en dispositivos de lectura biométricos, específicamente torniquetes.

9- Evitar una actualización de tarjetas (Físicamente).

10- Manejo de la ISO 14443.

11- El usuario interactuara con el sistema solo con su tarjeta.

12- Mostrar físicamente el estado del sistema (Activo o Inactivo).

Dados los requerimientos anteriores, se generaron los siguientes casos de uso de sistema.

Seguridad en tarjetas RFID

Tabla 1 Caso de uso N° 1

Caso de Uso	**Lectura de carnés**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda , Camilo Triana
Descripción	Se hace un lectura de la tarjeta mediante el módulo MFRC522
Actores	Usuario, Administrador.
Tipo de casos de uso	Normal
Precondiciones	El usuario debe poseer carné de la Universidad.
Curso normal	**Curso alternativo**
1. Se solicita el carné del usuario.	
2. El administrador debe operar el módulo MFRC522.	
3. Se realiza la lectura del carné.	**3.1. No se realiza una lectura completa de la tarjeta u ocurre un fallo de la misma.**
4. Se realiza el estudio de la arquitectura del carné, por bloques y por sectores. Además, de la búsqueda del Access Key.	**4.1. La arquitectura quedo corrupta por fallo en la lectura.**
Post condiciones: Si se encuentra las Access Key se dará acceso al usuario.	

Tabla 2 Caso de uso N° 2

Caso de Uso	**Escritura de carnés**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda , Camilo Triana
Descripción	Se realiza una escritura sobre los bloques y sectores de la tarjeta con el algoritmo planteado.
Actores	Administrador, Bases de datos
Tipo de casos de uso	Include
Precondiciones	El carné debe estar registrado y debe pasar por el proceso de lectura.

Curso normal	Curso alternativo
1.Se realiza la lectura del carné	**1.1. No se realiza una lectura completa de la tarjeta u ocurre un fallo de la misma.**
2. Se encuentra el primer sector del carné en el cual se encuentra las Access Key de ingreso y se sobrescribe con el algoritmo basado en la hora.	**2.1. No se encuentra el primer sector del carné.** **2.2. Ocurre un fallo en la edición del sector.** **2.3. No se realiza la sobrescritura.**
Post condiciones: Después de la edición del carné, el ingreso debe operar de manera correcta.	

Tabla 3 Caso de uso N° 3

Caso de Uso	**Editar permisos de carné**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	Se editan los permisos de ingreso según el cargo del usuario.
Actores	Usuario, Administrador.
Tipo de casos de uso	Extends
Precondiciones	Los usuarios deben estar registrados.
Curso normal	**Curso alternativo**
1. Se realiza la búsqueda del cargo del usuario.	**1.1. El usuario no postee un cargo actualmente.**
2. Se clasifica el usuario según su cargo (Estudiantes, Profesor, Administrativos, Oficios Varios).	**2.1. Se clasifican mal a los usuarios.** **2.2. No se clasifica al usuario en ningún cargo.**
3. Se implementaran permisos según el cargo Ej. Ascensor.	**3.1. Se les asignan permisos extras a los usuarios.** **3.2. Se deja al usuario sin ningún permiso de acceso**
Post condiciones: El usuario podrá acceder a las zonas en las que se le otorgaron permisos.	

Tabla 4 Caso de uso N° 4

Caso de Uso	**Actualizar algoritmo**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	Se estudian posibilidades para mejorar el algoritmo (Rendimiento).
Actores	Administrador.
Tipo de casos de uso	Extends.
Precondiciones	Se debe analizar el estado actual del algoritmo e para proponer mejoras.
Curso normal	**Curso alternativo**

1. Se planteas mejores de rendimiento al algoritmo.	
2. Se realizan las propuestas a la parte administrativa y de seguridad en la Universidad.	**2.1. La parte administrativa y de seguridad no aprueba las propuestas.**
3. Se trabajan en las mejores estipuladas y aprobadas.	**3.1. No se realiza el proceso de mejoras.** **3.2. No se invierten los recursos necesarios en las mejoras.**
Post condiciones: Se buscara la futura implementación de las mejoras (Largo o mediano plazo)	

Tabla 5 Caso de uso N° 5

Caso de Uso	**Implementar mejoras**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	Se implementaran las propuestas planteadas.
Actores	Administrador, Carnetización.
Tipo de casos de uso	Extends
Precondiciones	Las mejoras deben ser aprobadas.
Curso normal	**Curso alternativo**
1. Se revisara que las mejores sean compatibles con los sistemas actuales.	**1.1. Si no hay compatibilidad se plantea nuevas mejoras que si lo cumplan.**
2. Se implementaran las mejoras realizadas.	**2.1. Fallos en la implementación de las mejoras o archivos corruptos.**
3. Se analizara el uso pleno y óptimo de las mismas.	**3.1. Lecturas incompletas o fallos de escritura en los carnés.**
Post condiciones: Se lograra la plena implementación de las mejoras con mayor rendimiento del algoritmo.	

Tabla 6 Caso de uso N° 6

Caso de Uso	**Diagnóstico de errores**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	El administrador debe revisar los errores que presente el sistema
Actores	Administrador.
Tipo de casos de uso	Normal

Precondiciones	El sistema debe presentar fallos anteriormente.
Curso normal	**Curso alternativo**
1. EL administrador debe verificar el origen del fallo.	**1.1. No se encuentra el origen del fallo.**
2. Si el origen es físico se encargara la seguridad de la Universidad, pero si el fallo es de software será analizado por el Administrador.	**2.1. El error requiere de otras áreas.** **2.2. No se puede clasificar el error.**
3.Se hará un diagnóstico del fallo y se procederá con su respectiva reparación	**3.1. No se logra reparar el error.** **3.2. La reparación requiere demasiados recursos.**
Post condiciones: El sistema debe funcionar después del diagnóstico.	

Tabla 7 Caso de uso N° 7

Caso de Uso	**Registro de Usuario**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	Se registra el usuario en Carnetización.
Actores	Usuario, Carnetización, Bases de Datos
Tipo de casos de uso	Extends
Precondiciones	El usuario debe ser empleado o estudiante de la Universidad.
Curso normal	**Curso alternativo**
1. El usuario realiza envió de datos a Carnetización.	**1.1. El usuario no cumple con él envió de documentos.**
2. Se realizara el proceso de Carnetización del estudiante o empleado.	**2.1. No se genera un proceso de inscripción.** **2.2. El proceso de inscripción se diligencia de manera errónea.**
3. Se verificara el cumplimiento del pago de matrícula.	**3.1. Si no hay comprobante del pago de matrícula se cancelara el proceso.**
Post condiciones: El estudiante estará en la base de datos de Carnetización.	

Tabla 8 Caso de uso N° 8

Caso de Uso	**Entrega de Carnés solicitados**

Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana.
Descripción	Se hará entrega de los carnés solicitados por primera vez o cambios.
Actores	Usuario, Carnetización, Bases de Datos.
Tipo de casos de uso	Include
Precondiciones	El usuario debe procesar la solicitud o cambio de carné.

Curso normal	**Curso alternativo**
1.El usuario debe estar a paz y salvo o pagar el cobro de cambio de carné	**1.1.Si el usuario no está a paz y salvo o no ha pagado el valor del cambio no se hará entrega del carné**
2. Se debe dirigir a Admisiones donde se hará entrega del mismo además de probar su funcionamiento.	**2.1. La entrega del carné es errada.** **2.2. No se generan pruebas del carné por el personal de admisiones.**
Post condiciones: El estudiante tendrá su carné actualizado.	

Tabla 9 Caso de uso N° 9

Caso de Uso	**Generar reportes de usuarios**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	Se generaran reportes de los usuarios nuevos y antiguos
Actores	Usuario, Bases de Datos, Carnetización.
Tipo de casos de uso	Include
Precondiciones	Los usuarios deben estar registrados.

Curso normal	**Curso alternativo**
1.Se compila la información de los usuarios	**1.1. No hay datos guardados sobre los usuarios.**
2.Se analizara las actualizaciones y solicitudes de carné	**2.1. No se recogieron datos acerca de la actualización o solicitud de carnés.**
3. Los datos arrojadores serán estudiados para un control de Carnetización.	**3.1. Si los datos no son precisos no se realizara el proceso de análisis.**
Post condiciones: Los resultados del reporte ayudaran a medir la tasa de carnés dentro de la Universidad.	

Tabla 10 Caso de uso N° 10

Caso de Uso	**Mantenimiento de torniquetes**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	Revisión mensual del equipo físico de seguridad (Torniquetes).
Actores	Carnetización y seguridad.
Tipo de casos de uso	Normal
Precondiciones	Los equipos presentan fallas o desmejoras.
Curso normal	**Curso alternativo**
1.Se revisa el equipo para encontrar fallas o daños	**1.1. No se encuentran fallos.**
2. Se hace una corrección de dichos fallos en caso de que existan.	**2.1. Los fallos encontrados no pueden ser reparados.**
3. Se hará un mantenimiento de la parte estética en caso de ser necesario.	**3.1.No es necesario realizar mantenimiento de la parte este**
Post condiciones: Los torniquetes estarán en mejor estado y serán más óptimos.	

Tabla 11 Caso de uso N° 11

Caso de Uso	**Cambio dinámico de Access Key**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	Se realiza el cambio de Access Key guardadas en la base de datos.
Actores	Administrador, Bases de Datos.
Tipo de casos de uso	Include
Precondiciones	Los Access Key disponibles deben estar registrados en la base de datos
Curso normal	**Curso alternativo**
1. Se reconoce el Access Key que se está utilizando en ese momento.	**1.1. No se reconoce el Access Key.**
2.Se busca de manera aleatoria el nuevo Access Key que se implementara	**2.1. La búsqueda no arroja ningún resultado.** **2.2. El tiempo de búsqueda es muy extenso.**
3. Se asignara el Access Key encontrado al sector que corresponda a la hora del día.	**3.1. Se asigna el Access Key en un sector diferente.**
Post condiciones: Las Access Keys estarán en el sector indicado, listos para dar acceso.	

Tabla 12 Caso de uso N° 12

Caso de Uso	**Verificar existencia de Access Key.**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	Se verifica la existencia del Access Key de los carné
Actores	Usuario, Bases de datos
Tipo de casos de uso	Extends
Precondiciones	Los carnés deben estar registrados por Carnetización.
Curso normal	**Curso alternativo**
1. Se realiza la lectura de los carné.	**1.1. Fallo en la lectura.**
2. Se revisa la existencia del Access Key.	**2.1. No existe el Access Key.**
3. Se asigna un Access Key para próximas lecturas.	
Post condiciones: En cada lectura se realizara la verificación de Access Key.	

Tabla 13 Caso de uso N° 13

Caso de Uso	**Consultar Access Key**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	Se consulta la Access Key de un carné.
Actores	Usuario, Bases de Datos.
Tipo de casos de uso	Include
Precondiciones	Lectura previa de los carné.
Curso normal	**Curso alternativo**
1. Se revisa el sector asignado a la hora del día.	**1.1. Fallo en la lectura.**
2. Se revisa si el Access Key está asignado en este sector.	**2.1. El Access Key no está en este sector.**
3. Se compara con los Access Key de la base de datos, si coincide se dará acceso.	**3.1. En caso de que no haya una coincidencia se negara el acceso.**
Post condiciones: Las Access Key coincidirán con las registradas en la base de datos.	

Tabla 14 Caso de uso N° 14

Caso de Uso	**Verificación de Redundancia**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	Se hará una asignación del valor de las canciones según criterios de la casa disquera
Actores	Base de Datos.
Tipo de casos de uso	Include
Precondiciones	Lectura previa del carné
Curso normal	**Curso alternativo**
1.Se revisa los sectores del carné	**1.1. Fallo de lectura.**
2. Se revisa si el sector que escoge el algoritmo postee Access Key.	**2.1. Más sectores posteen Access Key.**
3. Se procederá a reestablecer los sectores que no sean indicados por el algoritmo.	
Post condiciones: Solo el sector asignado tendrá Access Key.	

Tabla 15 Caso de uso N° 15

Caso de Uso	**Accesos especiales**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	Accesos restringidos a ciertos usuarios.
Actores	Usuario, Bases de Datos, Administrador, Carnetización
Tipo de casos de uso	Extends
Precondiciones	El usuario debe estar registrado.
Curso normal	**Curso alternativo**
1. Se revisa el cargo que ocupa el usuario.	
2. Según su cargo se asignan accesos, como torniquetes especiales o ascensor.	**2.1.Se asignan roles a usuarios incorrectos**
Post condiciones: Cada usuario tendrá los accesos asignados según su cargo.	

Tabla 16 Caso de uso N° 16

Caso de Uso	**Realizar ingreso a la universidad**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	Entrar a la universidad por medio del carné
Actores	Usuario, Bases de datos
Tipo de casos de uso	Include
Precondiciones	El usuario debe portar carné de la Universidad.
Curso normal	**Curso alternativo**
1. El usuario colocara su carné en el torniquete para su debida lectura.	**1.1. Error en lectura.**
2. Si el Access Key es compatible se dará ingreso a la Universidad.	**2.1. Error en la verificación de Access Key**
Post condiciones: No se podrá registrar la misma canción	

Tabla 17 Caso de uso N° 17

Caso de Uso	**Solicitar cambio de carné**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	El usuario solicita un cambio de carné por un motivo específico.
Actores	Usuario, Carnetización.
Tipo de casos de uso	Extends
Precondiciones	El usuario debe estar registrado
Curso normal	**Curso alternativo**
1. Se debe realizar un solicitud de cambio de carné por escrito en donde se presente el motivo del cambio (Deterioro, Cambio de documento etc.)	
2. La solicitud debe ser radicada en Carnetización.	**2.1. Carnetización no se encuentra en servicio**
3. La solicitud es aprobada y se hará entrega del nuevo carné en días posteriores.	**3.1. La solicitud fue negada.**
Post condiciones: El carné se cambia y actualiza de manera correcta.	

Tabla 18 Caso de uso N° 18

Caso de Uso	**Actualizar carné**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda
Descripción	Se actualiza el carné debido a un cambio de cargo o ciclo académico.
Actores	Usuario, Bases de Datos, Carnetización.
Tipo de casos de uso	Normal
Precondiciones	El usuario debe estar registrado.
Curso normal	**Curso alternativo**
1.Presentar motivo de la actualización de carné	
2. Se comprueban los requisitos para el cambio, sea académico o de carácter profesional.	**2.1. Los requisitos exigidos no están aprobados por el usuario.**
3. Se entrega el carné al usuario con sus cambios correspondientes.	
4. Se actualiza los datos en la base de datos correspondiente.	**4.1. No se cambian los datos dentro del sistema.**
Post condiciones: El usuario se le entrega su carné actualizado.	

Tabla 19 Caso de uso N° 19

Caso de Uso	**Registrar carné**
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana
Descripción	Se registra el carné según los datos presentados en la inscripción o contratación.
Actores	Usuario, Bases de Datos, Carnetización.
Tipo de casos de uso	Include
Precondiciones	El usuario debe presentar proceso de inscripción o contratación.
Curso normal	**Curso alternativo**
1. **Se realiza el proceso de personalización de carné, según los datos del usuario.**	**1.1. El usuario suministra datos erróneos.**
2. Los datos se guardan en la base de datos.	**2.1. No se guardan de manera correcta los registros.**
3. Se asigna un tiempo de uso para carné generalmente es un semestre.	
Post condiciones: El carné está listo para su entrega.	

Tabla 20 Caso de uso N° 20

Caso de Uso	**Adquirir carné**	
Autor-Fecha	Diego Rodríguez, Camilo Cepeda, Camilo Triana	
Descripción	El usuario recibe su carné cuando esté aprobada su inscripción o contratación.	
Actores	Bases de Datos, Usuarios, Carnetización.	
Tipo de casos de uso	Include	
Precondiciones	El usuario debe aprobar su proceso de inscripción o contratación.	
Curso normal		**Curso alternativo**
1. El usuario debe presentar su recibo de matrícula u orden de contratación.		**1.1.**
2. Se realiza junto al usuario la verificación de datos.		**2.1. Los datos son erróneos,**
3. Se hace entrega del carné al nuevo usuario de la Universidad.		
Post condiciones: El usuario postee su carné que lo acredita como usuario de la Universidad.		

Fuente: Los autores

A continuación, se presentarán los diagramas de diseño de software, en los cuales se muestra la organización, los recursos y el funcionamiento del proyecto, especialmente del algoritmo de asignación de Access Key. Gracias a los diagramas, la estructura y la programación del proyecto es más clara, alcanzando una codificación más eficaz y que cumpla con los requerimientos iniciales del proyecto.

Cada una de las tareas principales del algoritmo se ven representadas en los diagramas, sustentando de manera claro cómo es su funcionamiento, su conexión con bases de datos y como es la estructura en el flujo de datos y recursos.

En primer lugar, se tiene los Diagramas de componentes, que representa como esta dividió el sistema en componentes y muestra la dependencia de dichos componentes.

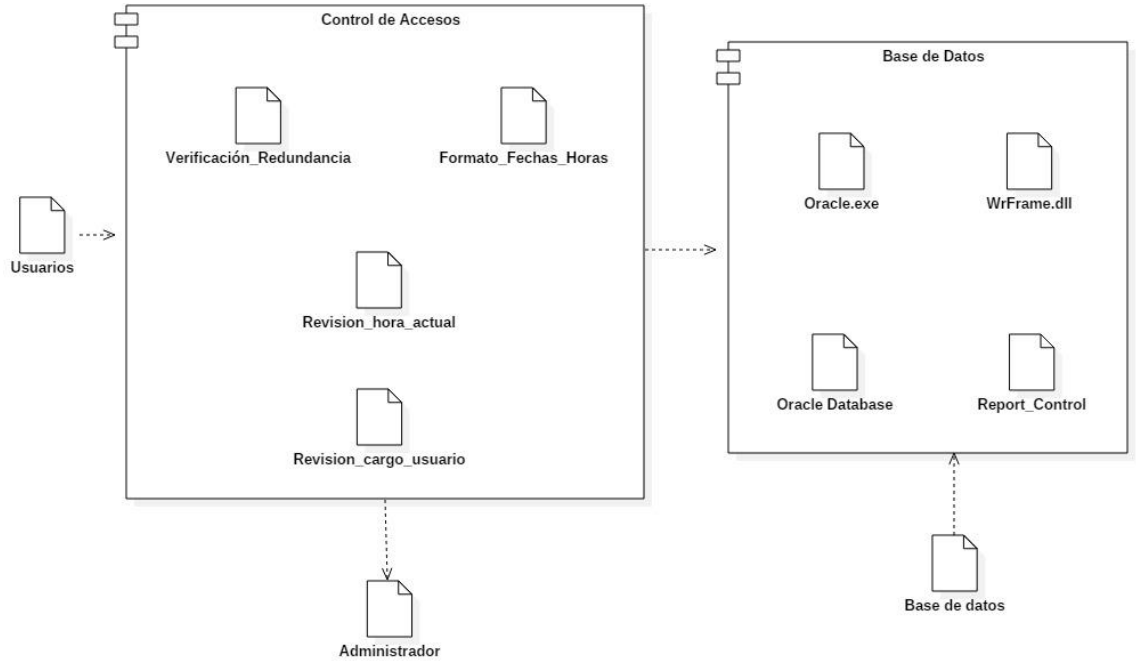


Ilustración 1 Diagrama de componentes Control de Accesos

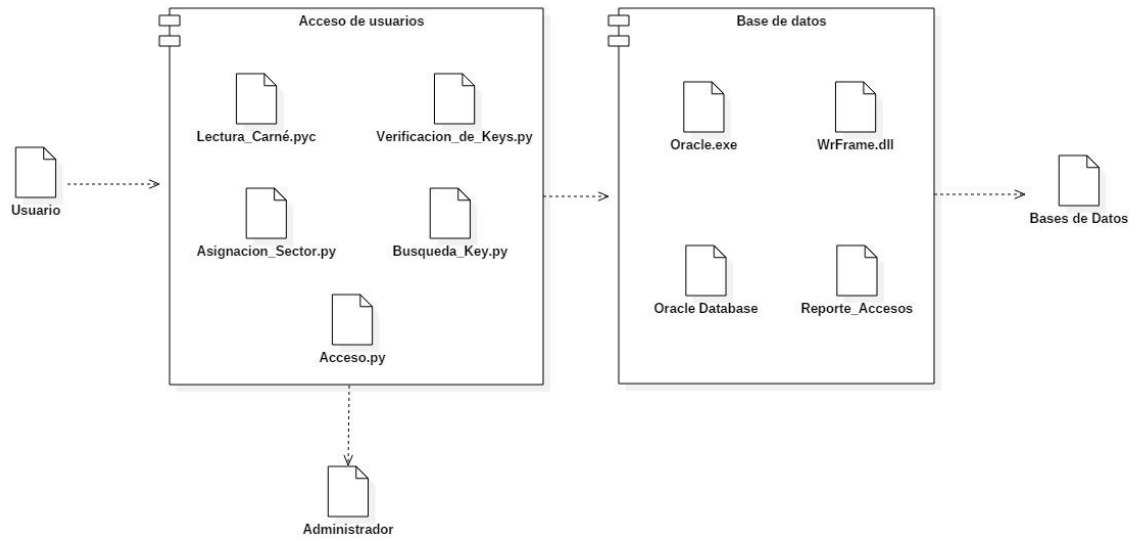


Ilustración 2 Diagrama de componentes Accesos de usuarios.

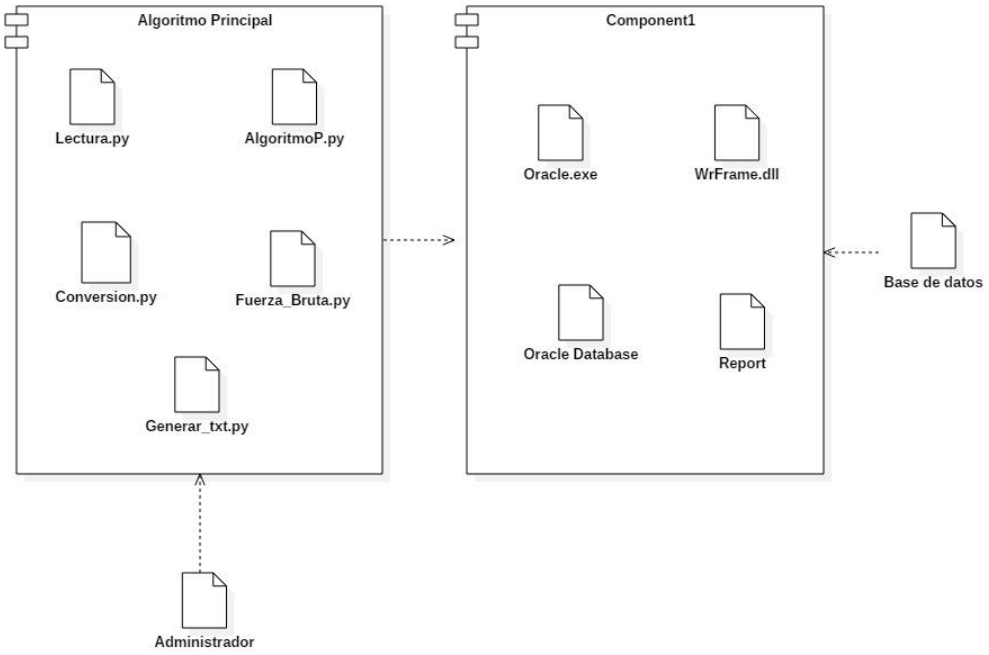


Ilustración 3 Diagrama de componentes Algoritmo

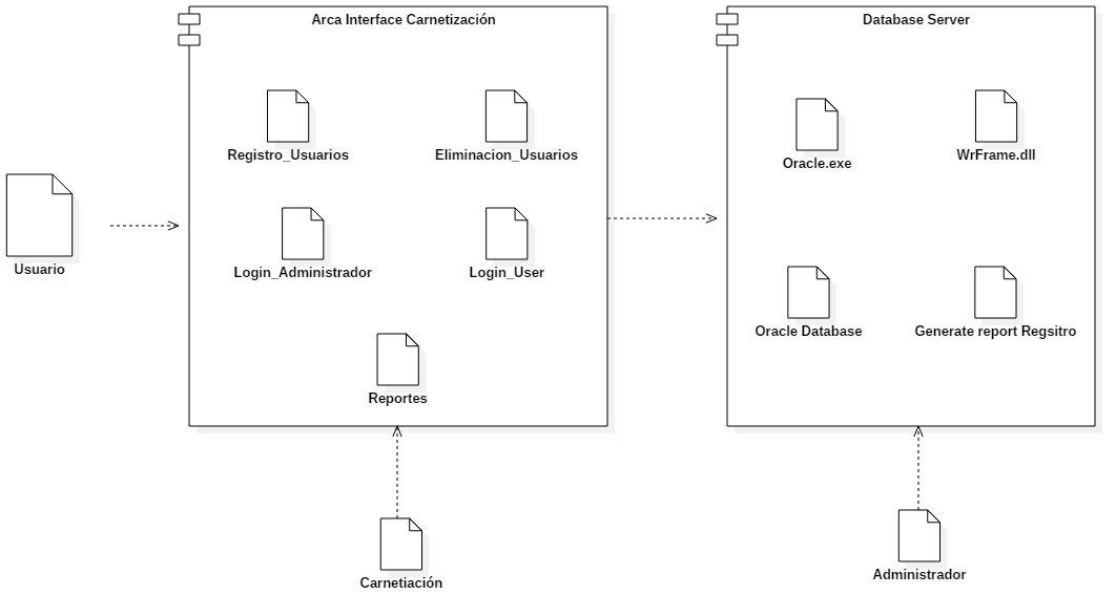


Ilustración 4 Diagrama de componentes de Registros

Los Diagramas de frecuencia los usamos para mostrar cómo es la interacción de los objetos y de la información que proveen las tarjetas RFID.

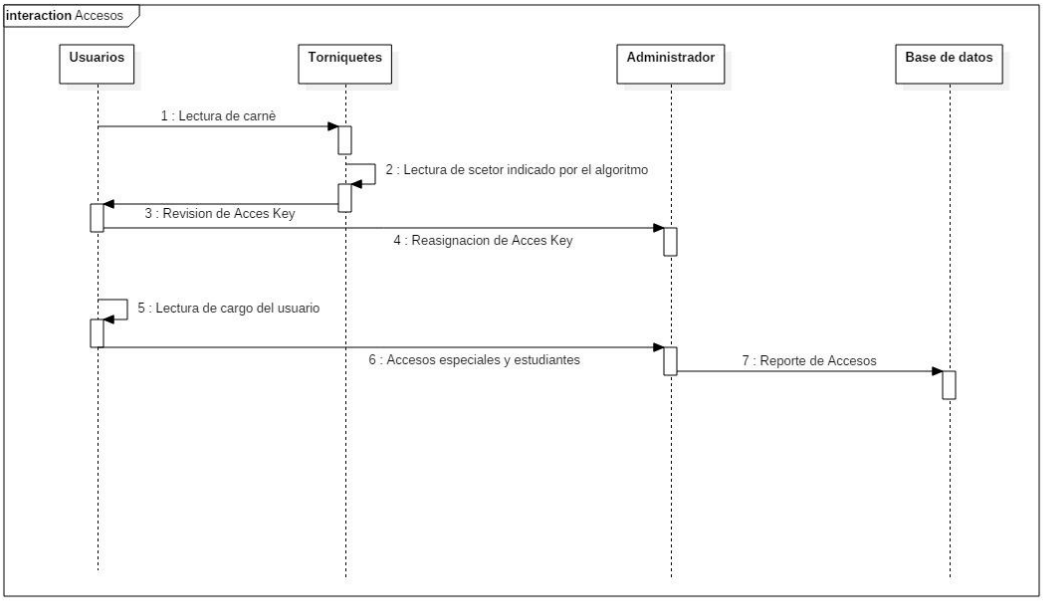


Ilustración 5 Diagrama de secuencia de Accesos

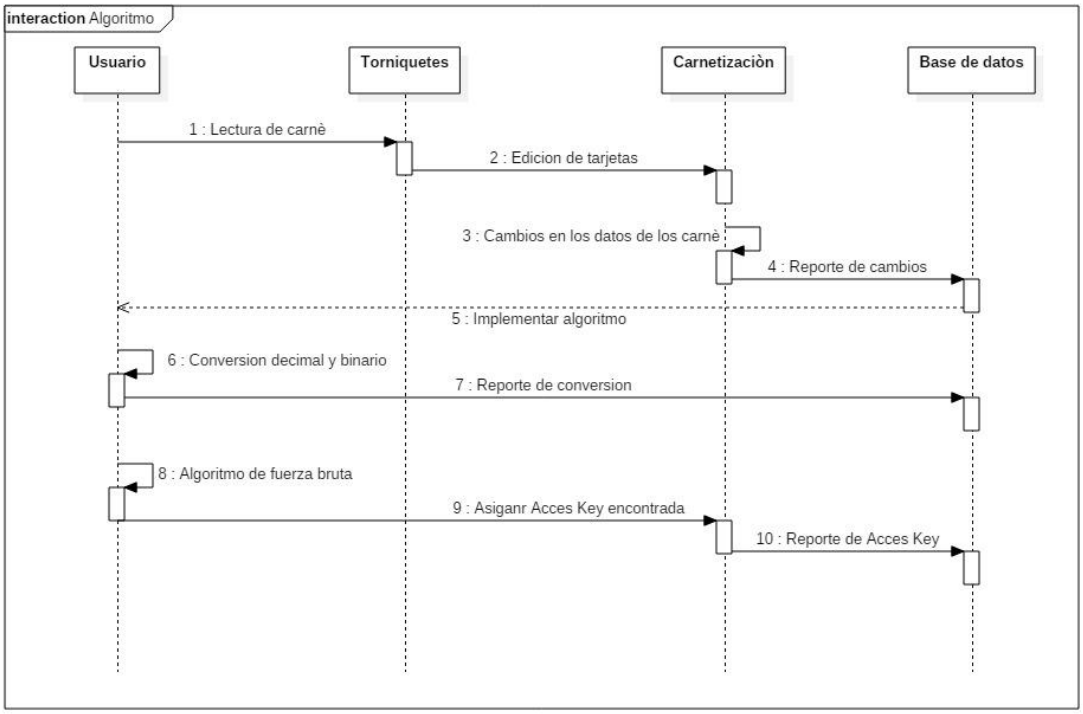


Ilustración 6 Diagrama de secuencia de Algoritmo

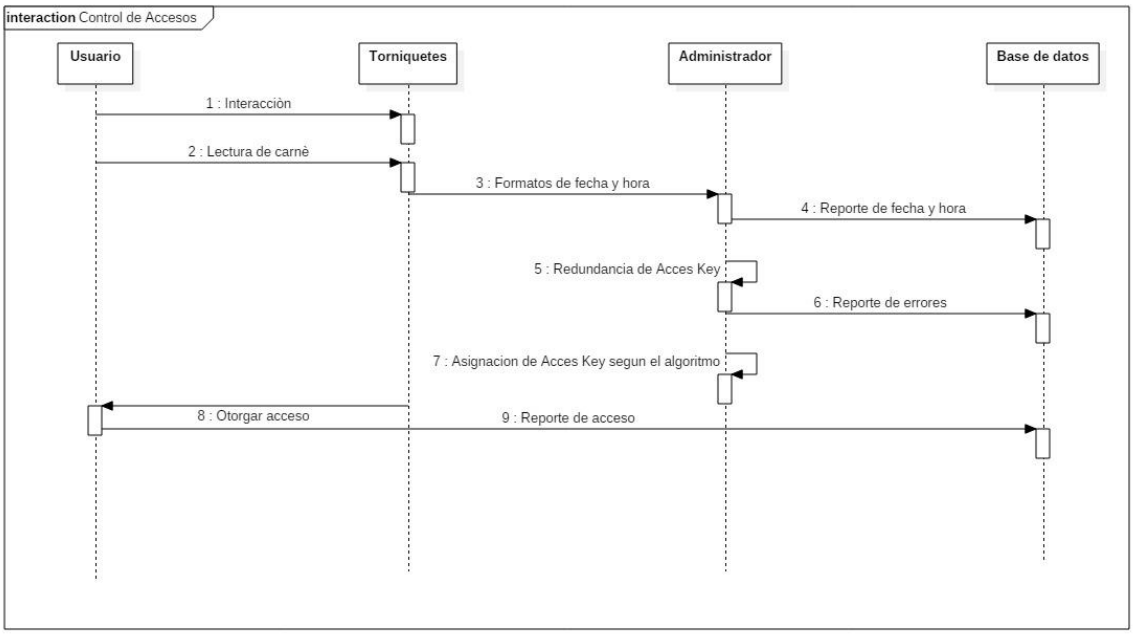


Ilustración 7 Diagrama de secuencia de Control de Accesos

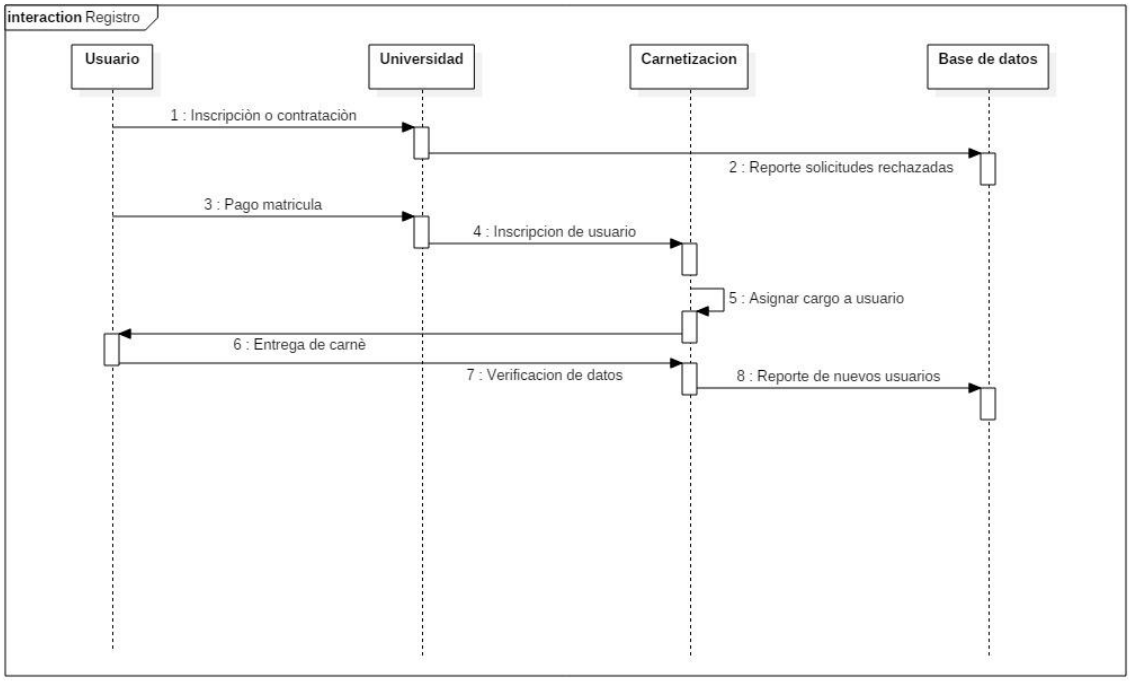


Ilustración 8 Diagrama de secuencia de Registro

El Diagrama de casos de uso demuestra la naturaleza de los casos de uso, su funcionamiento dentro del sistema y de cada objeto que interactué con dicho sistema.

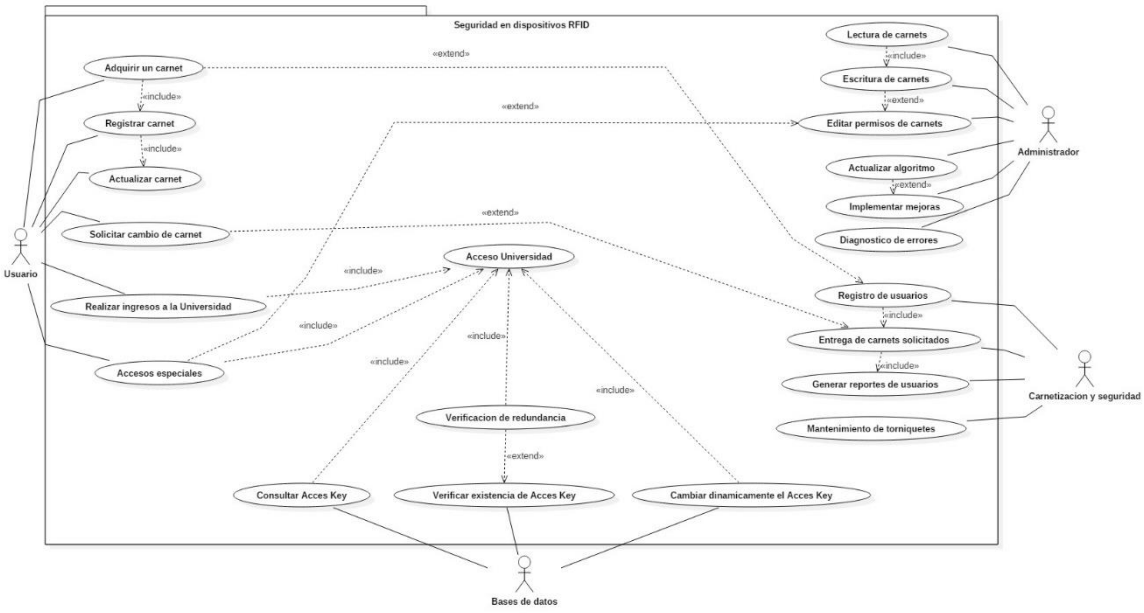


Ilustración 9 Diagrama de casos de uso

El Diagrama de avtividades demuestra la funcionalidad y las actividades que se pueden presentar durante el funcionaiento del software.

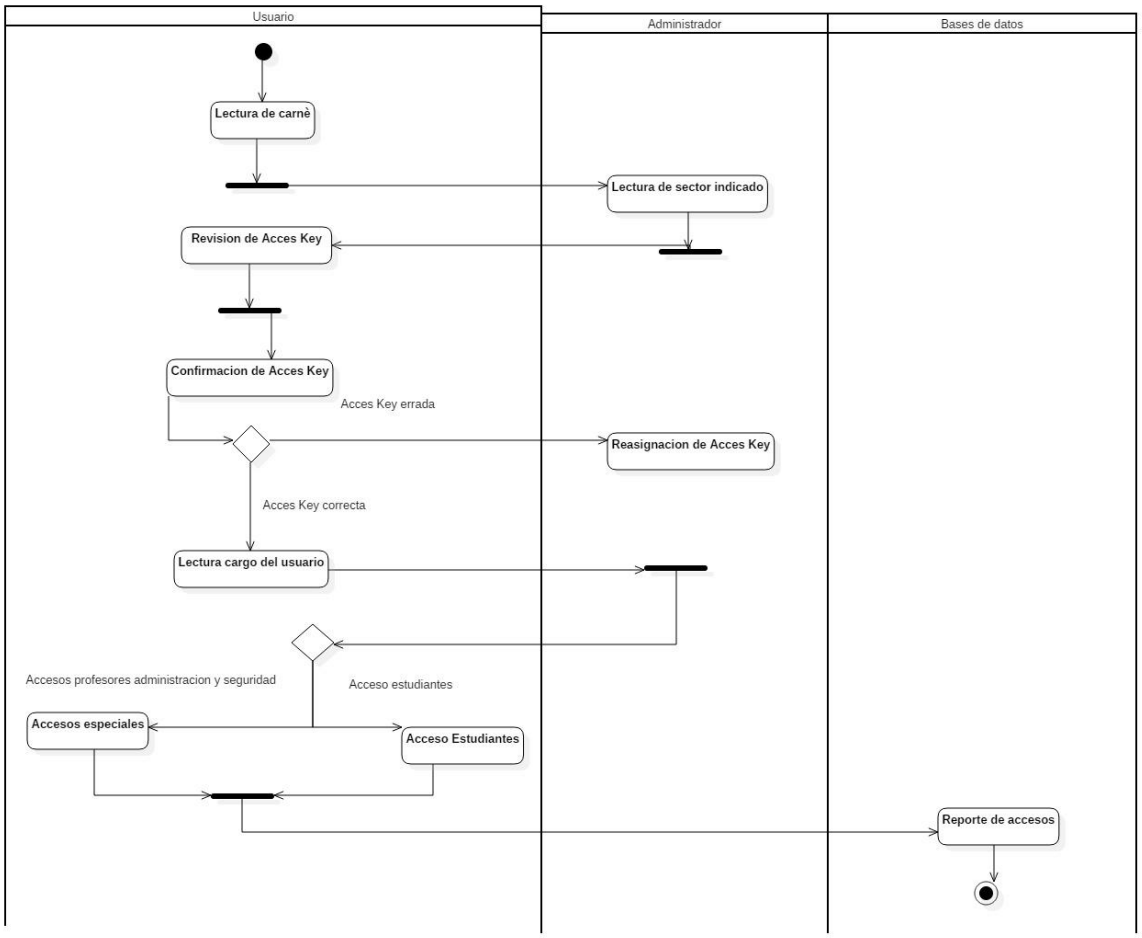


Ilustración 10 Diagrama de actividades de accesos

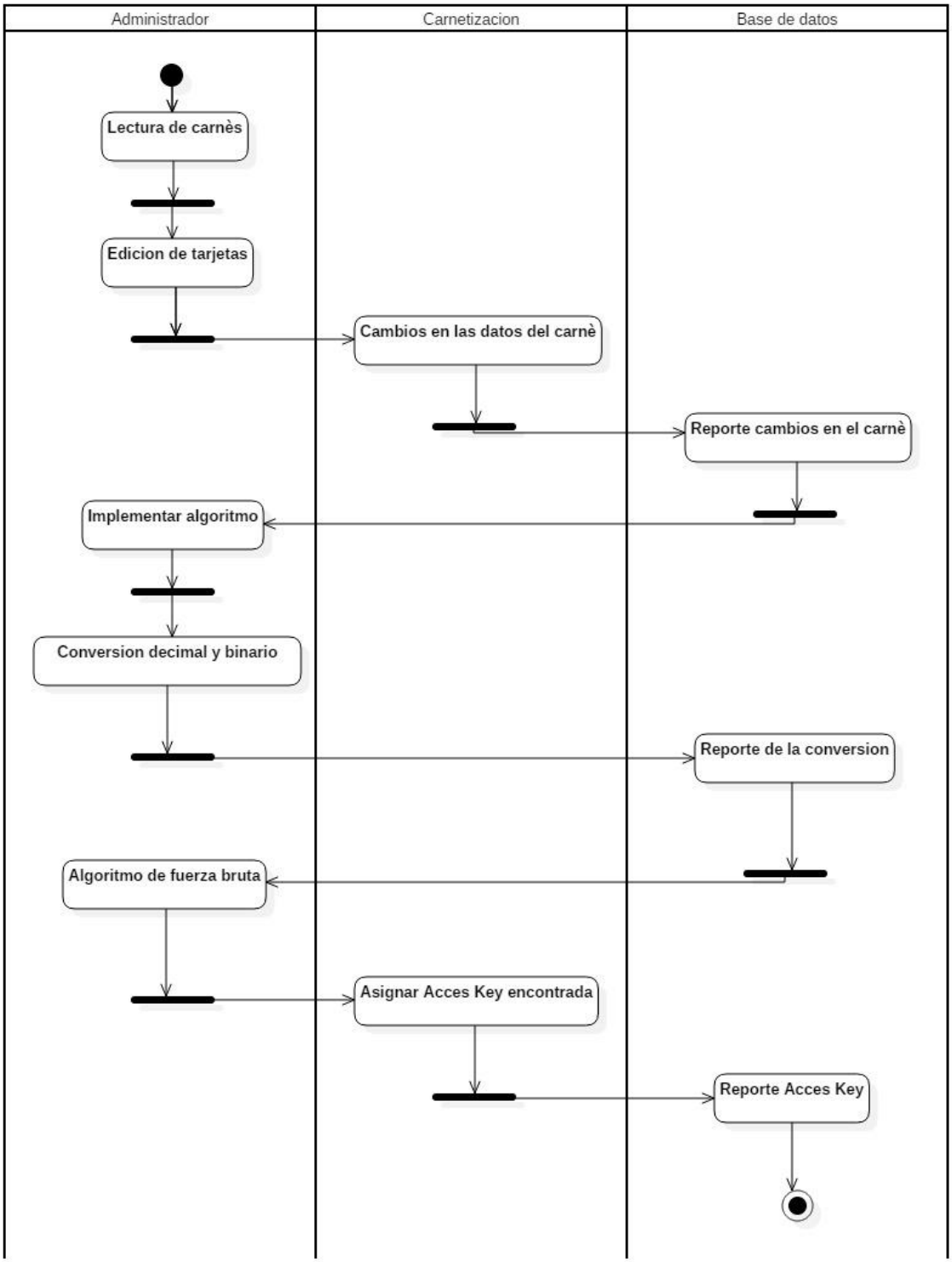


Ilustración 11 Diagrama de actividades de Algoritmo

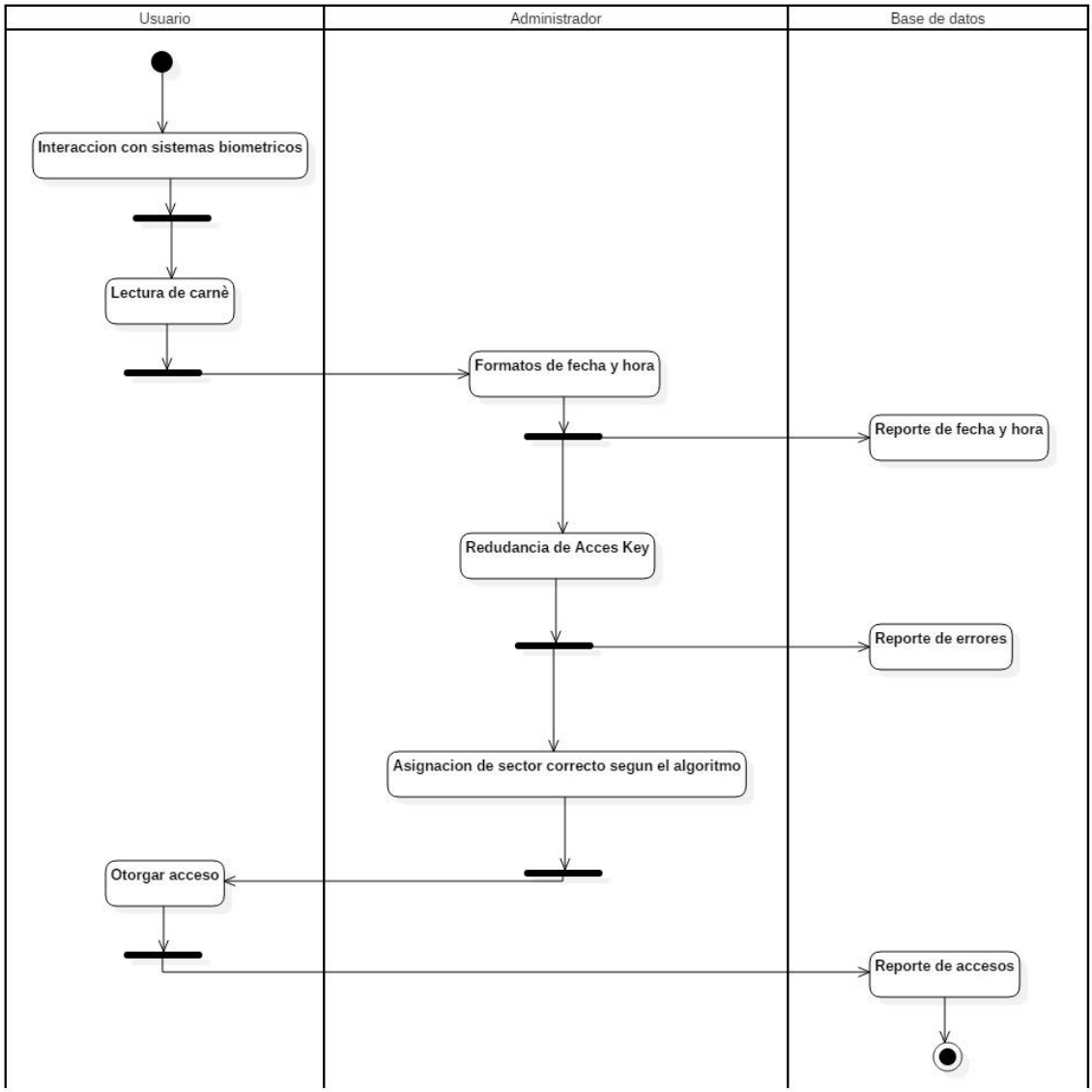


Ilustración 12 Diagrama de actividades de acceso

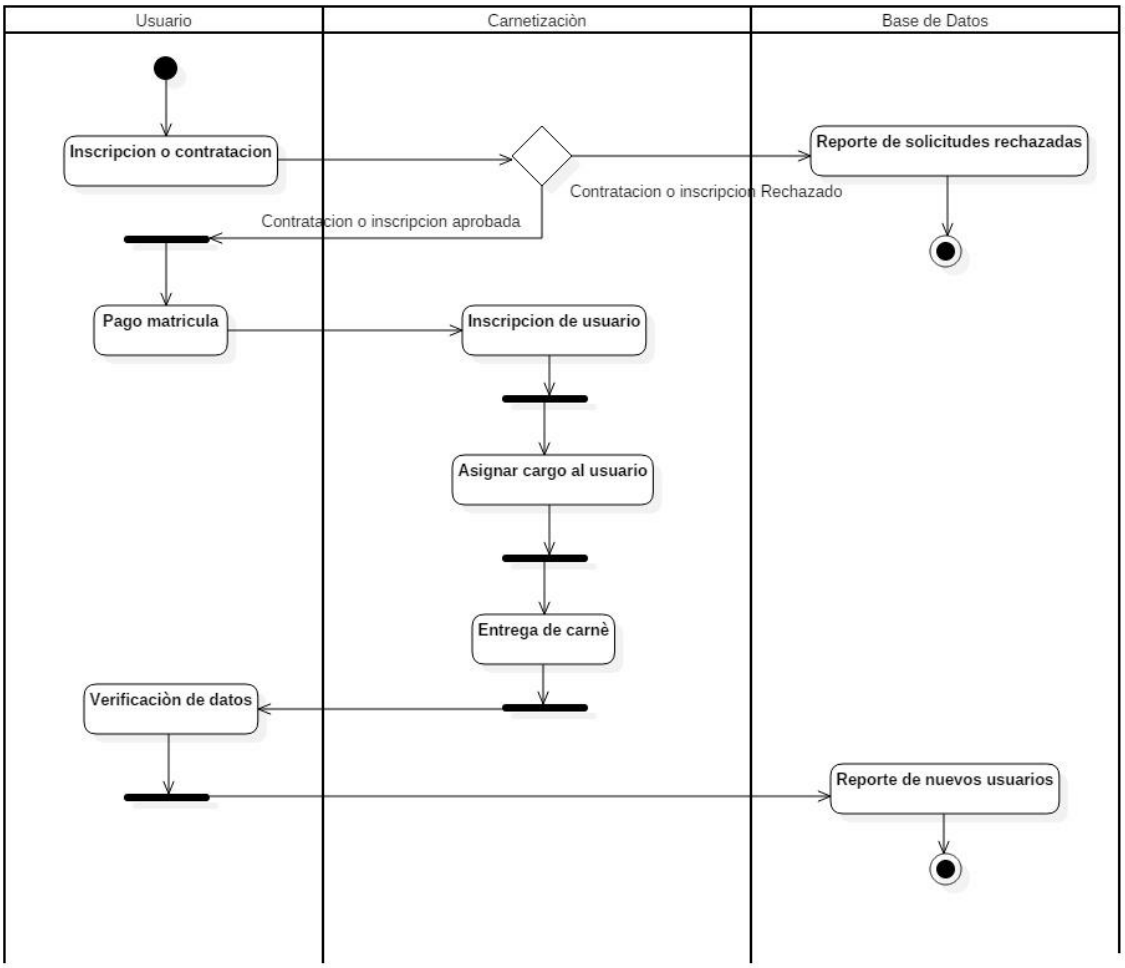


Ilustración 13 Diagrama de actividades de registro

Fase de planeación.

Como primera actividad en el proyecto después del debido proceso de recolección de información y levantamiento de requerimientos, está la realización del algoritmo, basados en los estándares existentes y su respectivo diseño hasta lograr el prototipo semi-funcional.

Como recursos a utilizar dentro del proyecto se estima el uso de componentes básicos tales como un computador que cuente con el lenguaje Python y su respectivo IDE[2], además de un dispositivo para realizar pruebas, para este objetivo se utilizó Raspberry Pi 2 con el módulo MRFC522 que realiza el proceso de lectura y escritura de dispositivos RFID tal como lo hacen los torniquetes actualmente implementado en la Universidad.

Por costos en el diseño actualmente presentado, solo se requiere el dispositivo Raspberry y su respectivo módulo con un estimado de $175.000.

En términos de tiempo se estima que todo el proceso de requerimientos y diseño del algoritmo y su respectivo documento se realizará en un aproximado de 6 meses.

En un trabajo que tiene como énfasis la realización de un sistema de seguridad, además de una curva de aprendizaje amplia dentro del campo de la electrónica y dispositivos a bajo costo trabajados durante estos 6 meses en el semillero en el cual el proyecto inició.

Alcance propuesto

El proyecto comprende varias instancias dentro de la Universidad. Se involucrará a toda la comunidad académica y administrativa de la institución, se emplearán diversos recursos dentro del proyecto entre ellos módulos electrónicos como Raspberry Pi y MFRC522 que realizarán el papel de un emulador de un torniquete. En cuestión de tiempo se utilizará un rango de 7 a 8 meses aproximadamente. El proyecto tiene aspectos de IoT por lo que es un proyecto que se abalara como un proceso que tiene como base principal un semillero.

[2] Entorno de desarrollo integrado (IDE): https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado

Análisis de resultados

Luego de investigar los diferentes tipos de dispositivos microcontroladores que existen en el mercado, además de realizar pruebas de rendimiento de los mismos se tomó la decisión de usar una placa Raspberry PI 2 modelo B (ver Ilustración 14) debido a que cumple con todos los requerimientos necesarios para el control de los periféricos, su alta capacidad de programación y su autonomía de trabajo, la cual cuenta con las características necesarias para mantener en ejecución un sistema operativo Debian (seleccionado luego de la realización de pruebas con múltiples sistemas operativos), llamado Raspbian. Es necesario tener en cuenta que estos dispositivos usan un sistema de procesador llamado ARM[3], que son especializados en reducción de espacio y reducción de registros internos. La placa presenta pines electrónicos capaces de emular diferentes puertos lógicos, tales como seriales, paralelos y comunicación y de alimentación eléctrica, Raspbian es una distribución del sistema operativo GNU/Linux y por lo tanto libre basado en Debian Jessie para la Raspberry Pi.

Ilustración 14. Raspberry Pi 2 Modelo B

Fuente: https://www.raspberrypi.org/blog/raspberry-pi-2-on-sale/

[3] Arquitectura de procesadores ARM https://es.wikipedia.org/wiki/Arquitectura_ARM

Técnicamente el sistema operativo es una adaptación no oficial de Debian para ARM de Raspberry Pi, con soporte optimizado para cálculos en coma flotante por hardware, lo que permite dar más rendimiento en diversos casos.

Destaca también el menú "raspi-config" que permite configurar el sistema operativo sin tener que modificar archivos de configuración manualmente. Entre sus funciones, permite expandir la partición raíz para que ocupe toda la tarjeta de memoria, luego de cualquier instalación de un sistema operativo Linux en una SD se debe expandir el sistema de archivos., configurar el teclado, aplicar overclock, activación de los diferentes puertos serial que la placa tiene, acceso de usuarios por medio de SSH, entre otros.

Al ser una distribución de GNU/Linux las posibilidades son infinitas tales como automatización de apertura de puertas, lectura de variables biométricas y medioambientales entre otros. Todo software de código abierto puede ser recompilado en la propia Raspberry Pi para arquitectura ARM que pueda ser utilizado en el propio dispositivo en caso de que el desarrollador no proporcione una versión ya compilada para esta arquitectura. En esta distribución, como la mayoría, contiene repositorios donde el usuario puede descargar multitud de programas como si se tratase de una distribución para equipos de escritorio la mayoría de estos aportados por la comunidad. Todo esto hace de Raspberry Pi un dispositivo que además de servir como placa con microprocesador clásica, tenga la funcionalidad de un ordenador personal, convirtiéndolo en una alternativa, especialmente para personas con pocos recursos, para la extensión e incorporación de la informática en países en vía de desarrollo para aplicaciones que no soliciten muchos requerimientos de hardware y software, todo esto permite un acercamiento efectivo al termino IoT y TIC.

La Raspberry contiene puestos análogos y digitales en la parte superior de la placa llamados GPIO, los pines GPIO no tienen ningún propósito especial definido, y no se utilizan de forma predeterminada ya que pueden variar su utilidad dependiendo de la finalidad con la cual se va a utilizar.

GPIO se utilizan en:

- chips con pocos pines.
- chips multifunción: gestores de energía, códecs de audio, tarjetas de video.
- aplicaciones embebidas hacen un uso intensivo de GPIO para la lectura de varios sensores ambientales (IR, de vídeo, la temperatura, la orientación de 3 ejes, aceleración), y para enviar la salida a motores de corriente continua, audio, LCD o pantallas LED para el estado.

GPIO puede incluir:

- Pines GPIO que pueden ser configurados para ser entrada o salida.
- Pines GPIO que pueden ser activados / desactivados.
- valores de entrada se pueden leer (por lo general alto = 1, bajo = 0)
- valores de salida de lectura / escritura.
- valores de entrada que a menudo se pueden utilizar como IRQ [4](Interrupciones del procesador)

No necesariamente se necesita una pantalla conectada como periférico para obtener los datos de la Raspberry, gracias a su conectividad de red podemos acceder a ella desde cualquier terminal que permita el uso del protocolo SSH y la instalación de alguno de los diversos clientes que este postee, además para mostrar el entorno grafico de cada programa que usemos en la placa se usa un emulador de protocolo X11 llamado Xming[5], que interpreta los datos gráficos del host y los reconstruye en la terminal desde la cual se conectó.

La primera tarea que se planteó automatizar fue la apertura de puertas por medio de tarjetas, las cuales usan sistemas RFID-NFC[6], por lo que un sensor de esta categoría debía ser instalado, Linux tienen una gran afinidad con los diferentes lenguajes de programación que

[4] IRQ (interrupción) https://es.wikipedia.org/wiki/Interrupci%C3%B3n
[5] Xming http://www.straightrunning.com/XmingNotes/
[6] Identificación por radiofrecuencia- Comunicación de campo cercano http://www.dipolerfid.es/es/tecnologia-RFID

existen, en este caso Python[7] es la mejor opción para la instalación de software y controladores de dispositivos periféricos, se tomó la decisión de usar este lenguaje, acompañado de una placa de lectura MFRC-522 (ver Ilustración 15).

Ilustración 15. Módulo MFRC522

Fuente: http://www.hotmcu.com/mifare-1356mhz-rc522-rfid-card-reader-module-p-84.html

Al terminar las conexiones eléctricas necesarias entre el lector y el ordenador Raspberry, se procedió a crear el controlador, basado en lenguaje Python y su uso a bajo nivel en el sistema operativo. Cabe aclarar que esta capacidad no viene por defecto en el sistema, por lo que debe ser programado cada pin de entrada o salida que se vaya a utilizar, el uso de este lenguaje viene gracias a sus ventajas a comparación al resto, su uso versátil y dinámico de la memoria RAM del dispositivo, además de la gran velocidad de procesamiento ya que este no es un lenguaje compilado sino interpretado.

Una vez que se acerca la tarjeta a un lector, ésta se activa e inicia un proceso de intercambio con el lector para establecer una comunicación cifrada usando el algoritmo Crypto-1[8]. Este proceso es igual con todas las tarjetas, este también contiene un sistema de detección de colisiones y lectura múltiple que permite detectar cuando dos o más tarjetas se encuentran en el lector, informando que no se puede realizar la lectura.

Después de establecer un canal cifrado, la tarjeta envía un código de identificación de conexión, que usualmente es el número de serie de la tarjeta, aunque la norma ISO 14443 dice que este número puede ser aleatorio. Con este número de conexión el lector está en

[7] Python, ver marco teórico
[8] Algoritmo de encriptación Crypto-1 https://en.wikipedia.org/wiki/Crypto-1

capacidad de realizar cualquier operación en la tarjeta, previa presentación de las claves de acceso a los respectivos sectores.

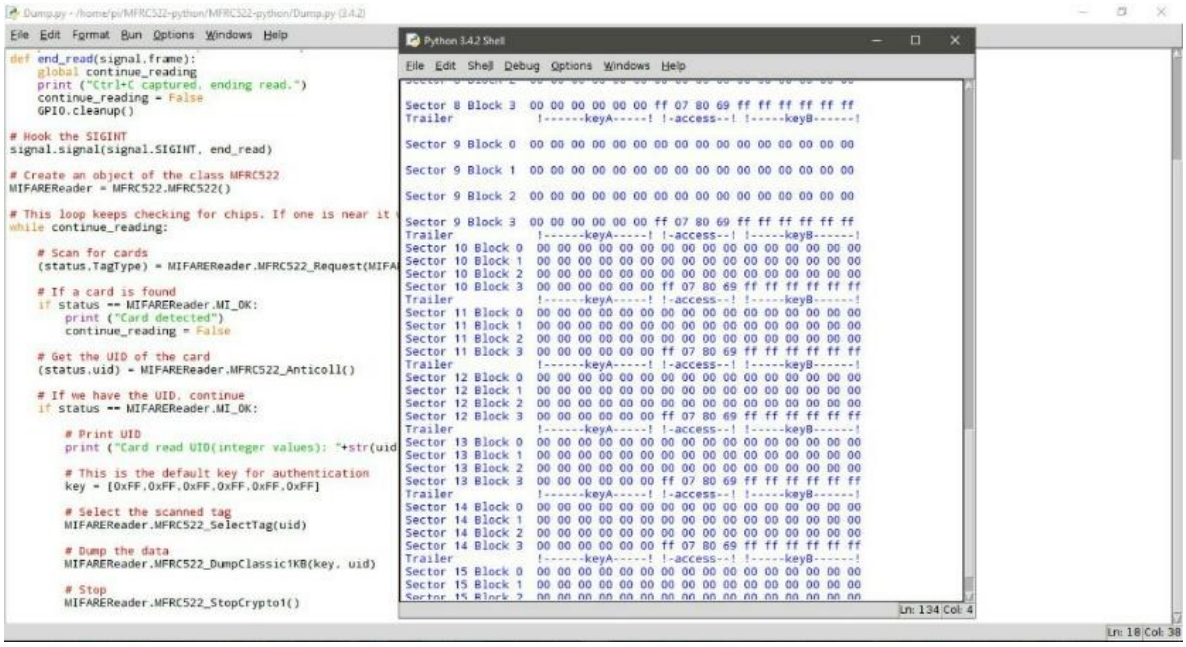
Después de la instalación previa del módulo mediante los puertos GPIO de la Raspberry, lo siguiente es utilizar Python para realizar las respectivas configuraciones para su pleno funcionamiento, se configura la seguridad de este para que otros dispositivos de radio frecuencia ajenos puedan establecer conexión. Además, los dispositivos RFID deben tener una capacidad de memoria para colocar nombres o información personal de los usuarios que utilizan el sistema.

El paso a seguir es realizar nuestra conexión a una base de datos MySQL previamente creada, en ella se guardan los registros que estén ligados al RFID, como una serie de datos personales. Con el fin de llevar un control de entradas y errores dentro del sistema, se debe tener en cuenta que dentro de nuestros carnets hay guardados una serie de keys editables que permiten el acceso. (Ver Ilustración 16)

Estos se conocen como primary key, campo principal de acceso a los carnets, sin este campo ninguna acción dentro del sistema sería válida, para efectos de permitir una conexión más estable entre el lenguaje de programación y la base de datos se usó el motor MySQL[9] aprovechando que tiene soporte para múltiples lenguajes.

Ilustración 16 Muestra de bloques de una tarjeta MIFARE

[9] Motor de bases de datos MySQL : https://www.mysql.com/



Fuente: Los Autores

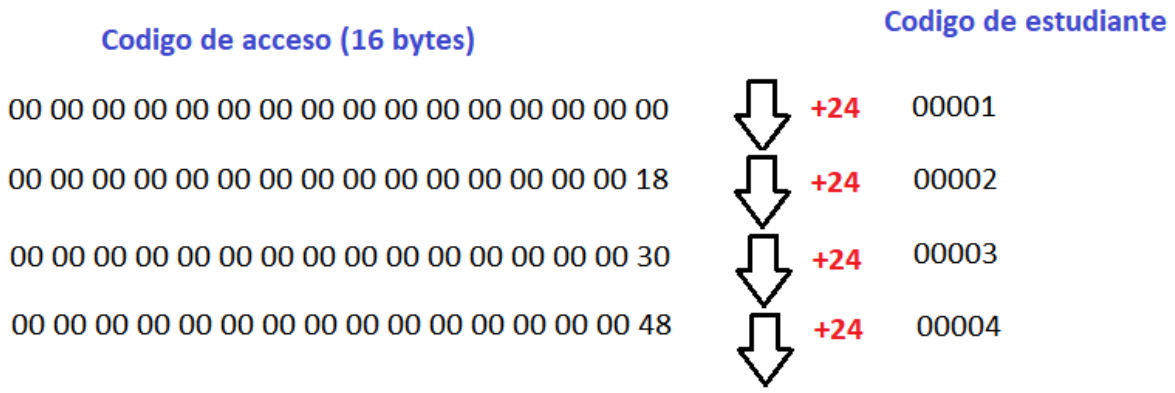
La directiva de seguridad que se plantea aplicar es el uso de los sectores de las tarjetas con respecto a las horas del día, por lo que se usaran 24 bloques de estas, por ejemplo si la hora es las 2 de la tarde, el software buscará y leerá el bloque lógico dentro de la tarjeta que se encuentre asignado para esta hora, si el código hexadecimal concuerda con un registro en la base de datos del personal se permitirá el acceso a las instalaciones, según la directiva cada persona en la base de datos debe tener asignado 24 códigos hexadecimales diferentes y únicos, y estos se deben encontrar escritos en la tarjeta carnet personal del usuario.

La cantidad de códigos hexadecimales diferentes que se pueden asignar son 4.294.967.294, lo que permite que cada usuario de tarjeta tenga 24 códigos propios solo para él, y la probabilidad de que haya una repetición por usuarios es mínima gracias a que el mismo software asignado busca los códigos ya asignados para verificar que los nuevos no sean iguales, para la asignación de estos se plantea un programa contador.

Para poder comprender lo expuesto, se ve en la ilustración 17 como se ingresan los códigos hexadecimales por código de un estudiante, el software de asignación selecciona los 24

códigos siguientes al último previamente asignado, el estudiante 00001 tiene los 24 primeros códigos, el 00002 los 24 siguientes al anterior, y el 00003 los 24 siguientes del ultimo asignado, de esta forma permitiendo que no ocurran repeticiones en la asignación.

Ilustración 17, asignación secuencial en valores hexadecimales.



Fuente: Los autores

Teniendo en cuenta el algoritmo presentado, la tabla 1 muestra cual bloque debe ser leído con respecto a la hora del día, además presenta los 24 códigos hexadecimales asignados al código estudiantil 00001, si son las 4 de la tarde el lector debe leer el bloque número 16, y buscar en la base de datos si el valor hexadecimal se encuentra ligado a un estudiante con estado “activo” dentro de la institución.

Tabla 21 Códigos vs Hora

Código:	**00001**		
Bloque	Valor hexadecimal (16 Bytes)	Valor Decimal (Ultimo Byte)	Hora del día
1	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0	1:00:00 a. m.
2	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01	1	2:00:00 a. m.
3	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	2	3:00:00 a. m.

4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03	3	4:00:00 a. m.
5	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04	4	5:00:00 a. m.
6	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 05	5	6:00:00 a. m.
7	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 06	6	7:00:00 a. m.
8	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 07	7	8:00:00 a. m.
9	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08	8	9:00:00 a. m.
10	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 09	9	10:00:00 a. m.
11	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0A	10	11:00:00 a. m.
12	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0B	11	12:00:00 p. m.
13	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0C	12	1:00:00 p. m.
14	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0D	13	2:00:00 p. m.
15	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0E	14	3:00:00 p. m.
16	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0F	15	4:00:00 p. m.
17	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 11	16	5:00:00 p. m.
18	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 12	17	6:00:00 p. m.
19	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 13	18	7:00:00 p. m.
20	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 14	19	8:00:00 p. m.
21	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 15	20	9:00:00 p. m.
22	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16	21	10:00:00 p. m.
23	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 17	22	11:00:00 p. m.
24	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	23	12:00:00 a. m.

	00 00 00 00 18		

Fuente: Los autores

Recursos

Recursos humanos.

Semillero Internet de las cosas (IoT) Coordinación de Ingeniera de Sistemas.

Ing. Alexander Sabogal Rueda - Docente

Diego Fernando Rodríguez Castañeda - Estudiante

Camilo Alberto Cepeda Mejía - Estudiante

Cristian Camilo Triana Redondo - Estudiante

Recursos institucionales.

Universidad ECCI:

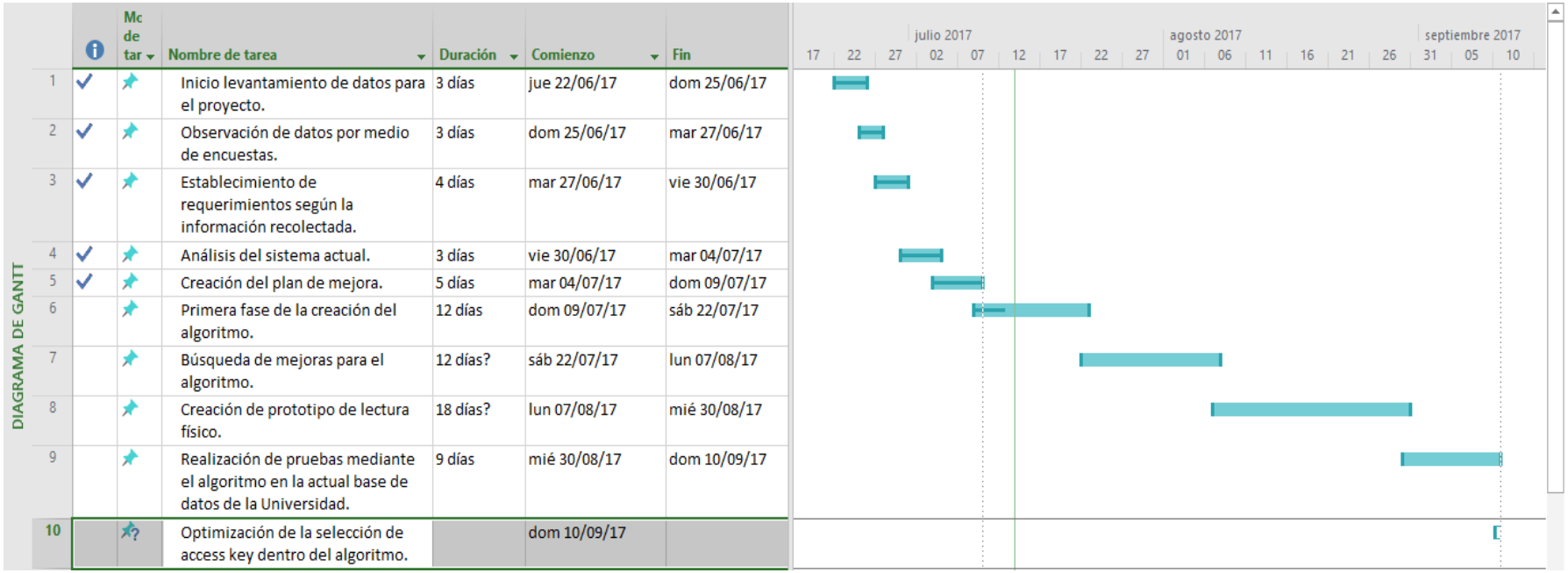
Área de registro y control

Área de seguridad

Coordinación Ingeniería de Sistemas / Tecnología en Desarrollo Informático.

Cronograma

Tabla 22 Cronograma de Actividades



	Mc de tar	Nombre de tarea	Duración	Comienzo	Fin
1	✓	Inicio levantamiento de datos para el proyecto.	3 días	jue 22/06/17	dom 25/06/17
2	✓	Observación de datos por medio de encuestas.	3 días	dom 25/06/17	mar 27/06/17
3	✓	Establecimiento de requerimientos según la información recolectada.	4 días	mar 27/06/17	vie 30/06/17
4	✓	Análisis del sistema actual.	3 días	vie 30/06/17	mar 04/07/17
5	✓	Creación del plan de mejora.	5 días	mar 04/07/17	dom 09/07/17
6		Primera fase de la creación del algoritmo.	12 días	dom 09/07/17	sáb 22/07/17
7		Búsqueda de mejoras para el algoritmo.	12 días?	sáb 22/07/17	lun 07/08/17
8		Creación de prototipo de lectura físico.	18 días?	lun 07/08/17	mié 30/08/17
9		Realización de pruebas mediante el algoritmo en la actual base de datos de la Universidad.	9 días	mié 30/08/17	dom 10/09/17
10		Optimización de la selección de access key dentro del algoritmo.		dom 10/09/17	

Fuente: Los Autores

Bibliografía

Dennis, A. K. (2013). *Raspberry Pi Home Automation With Arduino* . UK: Packt Publishing Ltd.

Esteve, J. J. (2014). *Administracion de un sistem operativo GNU/Linux.* Barcelona: Oberta UOC Publishing.

Jeremy Jones, N. G. (2009). *Python for Unix and Linux System Administration.* Estados Unidos: O'Reilly Media.

Joe Andrew Salazar, L. M. (1998). *Estados Unidos Patente nº US5802467 A.*

Oz, E. (2008). *Administración de sistemas de información 5ª Edición.* Mexico: Cengage Learning Editores.

Singh, S. R. (Mayo de 2016). Enhancing Cloud Data Security with Data . *Scientific Journal Impact Factor* , págs. 191-196.