Universidad de las Ciencias Informáticas

Facultad 1

Título: Solución para gestionar el control de acceso a instalaciones utilizando tarjetas inteligentes

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores: Kirenia Garcia Pérez
Orén Fornaris Cabreja

Tutor: Ing. Katerina Pereda Viñolo
Cotutor: Ing. Vismar Fernández Santana

"Ciudad de La Habana, 2010"

Frase

"La soberanía del hombre está oculta en la dimensión de sus conocimientos."

Francis Bacon: (22 de enero de 1561 – 9 de abril de 1626), primer vizconde de San Albano, canciller de Inglaterra, fue un célebre filósofo, político, abogado y escritor. Es considerado el padre del empirismo. Sus obras y pensamientos ejercieron una influencia decisiva en el desarrollo del método científico.

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos al Centro de Identificación y Seguridad Digital de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

_____ Kirenia Garcia Pérez

_____ Orén Fornaris Cabreja

_____ Ing. Katerina Pereda Viñolo.

_____ Ing. Vismar Fernández Santana.

Agradecimientos

Generales: agradecemos a todas las personas que han colaborado con nosotros de alguna u otra manera, en especial a los integrantes del departamento de Tarjetas Inteligentes, tanto profesores como estudiantes. A nuestros tutores, tribunal y oponente, por la ayuda brindada. A la universidad y los profesores que nos formaron. A la Revolución.

Oren:

A mis padres por darme la vida y todo su amor, por estar siempre pendiente de mí, por ser los principales guías y educadores de mi vida, les agradezco siempre su apoyo incondicional, por ello, todo mi amor y respeto.

A mi familia que de una manera u otra han colaborado para que este sueño se haga realidad, y me han apoyado siempre, a mi tía Yurkis muchas gracias por todo el apoyo brindado.

A mis amigos y compañeros que hemos formado una pequeña familia y de una u otra forma me han ayudado a llegar aquí.

A mi tutores por habernos guiado y preparado para enfrentarnos a este trabajo.

A los compañeros del proyecto y profesores, en especial a Dayron y a Ander por toda la ayuda brindada.

A Edistio por toda la ayuda brindada, y por estar siempre dispuesto a cooperar con nosotros.

A mi compañera de tesis Kirenia, por ser tan paciente conmigo, por soportarme todo este tiempo que hemos pasado juntos, muchas gracias.

A todos muchas gracias.

Kirenia:

Agradezco a Dios por ser una persona tan afortunada y tener una familia maravillosa, por tener dos madres (mi mamá y mi tía Mayra) que han dado todo por mí, por todo el amor, la comprensión y la paciencia que han tenido conmigo, gracias a ustedes, por querer hacer de mí una persona con principios y por enseñarme a querer mucho a la familia.

A mis hermanos, que son muy importantes en mi vida, en especial a Arlety por ser un ejemplo a seguir y por darme esa sobrina que quiero tanto, gracias por ser maravillosa conmigo. Además a Andrés que es mi hermano de corazón, te adoro y te doy las gracias por estar siempre a mi lado, desde que nací eres de las cosas más importantes de mi vida, gracias también por esos niños lindos (Elizabeth y Enmanuel), por tu comprensión y tu confianza; por ser mi hermano, mi primo, mi papá y mi amigo.

A todos, mis tíos, tías, primos (especialmente Kenia) y a mis abuelos, gracias a todos por darme consejos, preocuparse por mí y quererme tanto, en especial a Aracelis, que no importa donde esté, siempre se preocupó por mí.

A mi amiga Denise, por comprenderme, quererme y soportarme tantos años, a su mamá Diana por ser otra madre para mí. A mi amigo David, que más que amigo siempre fue mi hermano.

A Juan, que a pesar de su seriedad siempre me ha querido y se ha preocupado por mí.

A Mabel por impulsarme a vivir y enseñarme mil cosas de la vida, por ser especial, cuenta conmigo siempre. A sus padres Isis y Cabada, les agradezco que me hayan tratado siempre como una hija y cuidado en todo momento, los voy a querer siempre.

A mis padrinos Laurita y Yuri, gracias por ser geniales y preocuparse siempre.

En todos los años de universidad he tenido la dicha de conocer a gente que me ha querido y ayudado mucho, a los cuales nunca olvidaré:

En primer lugar le agradezco a Yoel por cuidarme tanto, por salvarme incluso de mí misma, por preocuparse y ocuparse de mí siempre, en las buenas y en las malas. A su hermana Zenia por su apoyo. A Ana por quererme y ayudarme, como si fuera mi madre.

A mis amigos: Claudia, Julio, Tatiana, Yaneisi, Zumeta, Yadira OConnor por interesarse por mí siempre. A mis compañeros de aula y las muchachas del apartamento, por soportarme.

A Raylith por todos sus regaños para que estudiara y me esforzara más y por hacerme reír.

A Dayli por preocuparse por mí y brindarme su apoyo y conocimientos.

A Amanda por preocuparse por mí, por obligarme a estudiar, por ser especial y formar parte de mi vida.

A Dailenis por ser maravillosa, te agradezco todo tu apoyo y las horas compartidas, siempre estaré ahí para ti, como lo has hecho tú. A Yaniel y Vicente, por ser como hermanos conmigo, quererme, cuidarme y soportarme, los voy a querer siempre, espero tenerlos cerca toda la vida.

A Dayron y Ander por explicarnos y ayudarnos cuando lo necesitábamos. A mis tutores, por el apoyo brindado. Al tribunal por el apoyo que nos dieron, a Edistio, por ser tan profesional y ayudarnos, a Yandi por atendernos cuando hacía falta.

Por último, pero no menos importante a mi compañero de tesis Oren, por ser mi amigo todos estos años, pero sobre todo por soportarme durante la confección de la tesis, eres como un hermano para mí. Sigue siendo tan maravilloso como eres y cuenta conmigo siempre y dondequiera que esté.

Dedicatoria

Oren:

A mi mamá y a mi papá, porque todo lo que soy se lo debo a la enseñanza moral, intelectual y física que recibí de ustedes.

A mi hermano por estar ahí para mí.

A mi familia y amigos por confiar en mí.

Kirenia: A mi familia, porque a ellos les debo lo que soy y son quienes me inspiran en todo. En especial a mi mamá. Para ellos es este trabajo con todo el amor del mundo.

Resumen

Con el auge de las tecnologías, vigente en la actualidad y la creciente necesidad de mantener la seguridad en las instalaciones, se desarrollan múltiples tipos de sistemas de control de acceso, en dependencia de las necesidades de donde se encuentren instalados. Debido a lo difícil que se hace controlar el acceso de personal, cuando los bienes tanto físicos como lógicos necesitan ser altamente protegidos y además aumentan y varían en tiempo y espacio, a la posible suplantación de personal a la hora de la identificación y autenticación de las personas, es necesario el uso de las tarjetas inteligentes, las cuales brindan seguridad en cuanto a los datos que poseen, efectividad y posibilitan ventajas con respecto a otras tecnologías.

En el Centro de Identificación y Seguridad Digital (CISED), de la UCI, se estudian estándares y tecnologías de las tarjetas inteligentes, para llevar a cabo soluciones informáticas que utilicen las mismas, debido a los beneficios económicos y científicos que estas conllevan, se decidió crear una solución para gestionar el control de acceso a instalaciones mediante el uso de tarjetas inteligentes, por lo cual se desarrolló el análisis, diseño e implementación de componentes `applet` y `middleware` para gestionar la solución planteada.
En el presente trabajo se realizó un profundo estudio de las tarjetas inteligentes, como resultado del mismo, se desarrolló una solución para el control de acceso utilizando las mismas. El documento recoge los resultados de la investigación realizada, describiéndose las principales características de los sistemas analizados, la arquitectura y el diseño de la solución propuesto.

Nota: para las palabras técnicas se usó una fuente de letra distinta a la del documento.

Palabras clave: sistemas de control de acceso, tarjetas inteligentes, `middleware`, `applet`.

Índice

Índice de Figuras

INTRODUCCIÓN

Por control de acceso se entiende, la supervisión del flujo de personas a un lugar restringido, ya sea de forma manual o con la utilización de otros mecanismos. Desde que el ser humano sintió la necesidad de restringir la entrada a instalaciones, surgieron ideas de cómo controlar la seguridad de las mismas.[1] Primeramente se logró de forma manual o personal, cuando un agente encargado de la seguridad reconocía directamente a la persona que quería acceder al local; luego se comenzaron a utilizar distintivos identificativos del lugar, pero esta práctica no era confiable debido a su fácil falsificación.[2] Con el objetivo de automatizar este proceso, se fueron desarrollando sistemas informáticos que monitoreaban y controlaban el tráfico a través de puntos de acceso.

Los Sistemas de control de acceso son una aplicación útil para la seguridad de las empresas con gran cantidad de empleados. Registran el acceso de cada empleado a las diferentes zonas de la instalación, lo cual permite controlar si existe una infracción, ya sea robo o invasión de propiedad, así como la fecha y el responsable de la misma. Con la amplia evolución que han experimentado dichos sistemas en los últimos años, se ha atenuado en gran medida la preocupación de que personal no autorizado acceda a una instalación determinada.[3] Están compuestos por tres procesos esenciales, el primero de ellos es la identificación, en el cual la persona declara su identidad como usuario hábil para acceder a la instalación, el siguiente es la autenticación, proceso en donde la persona ofrece pruebas que permite a la solución verificar su identidad y por último tiene lugar el proceso de autorización, que consiste en dar a la persona los privilegios que tiene en la instalación.[4]

Los sistemas de control en su mayoría incorporan el uso de diferentes instrumentos o dispositivos como una tarjeta de identificación, teclados, escáneres de huellas digitales y otros tipos de tecnología. En el mundo existe una amplia gama de estos sistemas, entre los que se pueden mencionar los sistemas de vigilancia, que además requieren de personal calificado para este trabajo. Los sistemas para el reconocimiento de placas de matrícula que capturan, procesan, interpretan y graban imágenes. No menos usados y eficientes, son los lectores, tanto los de pared, muy útiles para supervisar el acceso a puertas o pasillos, como los de paso, que permiten controlar personas (u objetos) a su paso por determinados puntos (puertas electrónicas) a pocos metros de distancia. Además están los sistemas biométricos, que posibilitan verificar la identidad de las personas a través de patrones biométricos, entre los que

se destacan la lectura de huellas dactilares, de iris y reconocimiento facial. Un gran número de instalaciones a nivel mundial basan su seguridad en este tipo de sistemas, sin embargo existen algunos inconvenientes, por ejemplo si una persona es ciega o padece de cataratas, tiene los ojos de color oscuro, la iluminación no es adecuada o el ángulo en que se coloque la cabeza es incorrecto, se dificulta la lectura de iris. Una huella dactilar puede no ser eficiente debido a un daño físico, a diferencia de una firma, la cual puede ser modificada, tanto por factores controlables, como por psicológicos no intencionales. Debido a las diferencias señaladas, no existe un único sistema biométrico que sea capaz de satisfacer todas las necesidades.[5]

Los sistemas de control se están imponiendo en temas de seguridad por lo fáciles y prácticos que son a la hora de utilizarlos. En dependencia del tamaño de las instalaciones, de los recursos económicos disponibles y del personal vinculado a las mismas, los sistemas anteriormente planteados pueden ser o no factibles, por lo que se desarrollan diferentes alternativas para resolver este problema. Existen algunos casos en que la supervisión de la entrada/salida de un local o área, es controlada a través de credenciales, estas generalmente son tarjetas, que poseen información referente al portador.[6]

Las tarjetas inteligentes (`smartcards`), o tarjetas con circuito integrado, son similares en tamaño y forma a las tarjetas de crédito bancario, contienen circuitos integrados que permiten la ejecución de cierta lógica programada. Tienen la capacidad de almacenar, encriptar y procesar información de manera segura, tal como lo hace un ordenador convencional. Las tarjetas no contienen baterías, la energía es suministrada por los lectores de tarjetas.[7] La tarjeta inteligente es básicamente un `chip`, encapsulado en un rectángulo de PVC (Ploricloruro de Vinilo) o plástico. El `chip` que contiene dispone de unos contactos exteriores que son los que le permiten mantener una comunicación con él y de esta forma acceder a la información que contiene o grabar nueva información. Su pequeño formato hace que sea ideal como sistema de identificación personal. Además, su medida no está limitada por razones técnicas, sino por razones de estandarización, es decir, técnicamente se podrían utilizar tarjetas que fuesen la cuarta parte de las actuales.[8]

Debido al constante avance tecnológico que existe en el mundo, las empresas y corporaciones crecen cada día y necesitan sistemas de acceso que ofrezcan mayor seguridad. En la actualidad, se ha registrado una tendencia al uso de dispositivos inteligentes de identificación, entre los cuales se destacan las tarjetas inteligentes, dando paso a un modelo innovador de control de acceso, que logra procesamiento rápido, autenticación personal y mitigación de

riesgos. Dicho modelo constituye la base para un sistema de identificación segura que además de supervisar el acceso, es el responsable de la seguridad, que se necesita para proteger la instalación.

Las tarjetas inteligentes son aplicadas hoy en muchas ramas, servicios y procesos de la sociedad, se han vinculado también al control de acceso, ya sea físico o lógico, el cual fue presumiblemente su primera aplicación social. Dichas tarjetas están teniendo cada vez más aceptación como la credencial de preferencia para controlar el acceso. Las tarjetas de identificación inteligentes basadas en estándares pueden ser usadas para autenticar la identidad de una persona, determinar el nivel de acceso adecuado y admitir físicamente al portador de la tarjeta, a un servicio o establecimiento. Más de una aplicación de acceso puede ser realizada en una tarjeta única de identificación inteligente, permitiendo a los usuarios tener acceso a recursos físicos y lógicos sin la necesidad de portar múltiples credenciales. La seguridad puede cambiar dinámicamente los derechos de acceso, dependiendo del nivel de amenaza percibido, la hora del día o cualquier otro parámetro requerido. Su nivel de seguridad es alto, por lo que reducen la posibilidad de fraude, su uso es fácil y mantienen los datos del titular bajo total privacidad. Cada tarjeta almacena información sobre el portador, protegiendo la misma de posibles accesos no autorizados.[9]

Como parte de las acciones encaminadas al desarrollo de la informática y las comunicaciones en Cuba, el país ha promovido el estudio de la informática y la apertura de nuevas empresas dedicadas al desarrollo de software. Se planea la creación de un parque tecnológico en Cuba, lo cual constituiría un paso importante para lograr la inserción del país entre las más prestigiosas cadenas productivas globales, además de avanzar hacia fases de mayor complejidad tecnológica.

En el año 2008 se creó en la Universidad de las Ciencias Informáticas, el Centro de Identificación y Seguridad Digital, el cual está compuesto por varias líneas de investigación y desarrollo, una de las cuales especializa su trabajo en el área de las tarjetas inteligentes, fomentando la creación de soluciones informáticas relacionadas con las mismas. Dentro de este departamento surge la idea de desarrollar una solución que posibilite el control de acceso a instalaciones utilizando dichas tarjetas.

Las tarjetas inteligentes vendrían a potenciar en gran medida el proceso de autenticación en el control de acceso. Existen 3 patrones fundamentales para comprobar la verdadera identidad de una persona, mediante la presentación de diferentes aspectos, primeramente algo que solo la

persona es capaz de conocer, como por ejemplo una contraseña o la respuesta a una pregunta clave; en segundo lugar de un objeto que solo la persona posee, dígase una credencial y en tercer lugar una característica intrínseca de la persona, una propiedad que la hace única tal como las huellas dactilares o los rasgos faciales.[10] Una tarjeta inteligente engloba los 3 tipos básicos de autenticación, ya que hace función de credencial y su seguridad se incrementa exigiendo al portador el conocimiento de un número de identificación personal (PIN, por sus siglas en inglés) y por último la propia tarjeta almacena patrones biométricos comprobables para verificar la identidad.[11]

Partiendo de lo anteriormente abordado se manifiesta la siguiente **Situación Problémica**: con el crecimiento tecnológico, hoy en día existe la necesidad en algunas instalaciones de contar con un sistema que posibilite controlar el acceso de personal a las mismas, mediante el uso de tarjetas inteligentes, con el objetivo de lograr un aumento en la seguridad y automatizar los procesos de identificación y autenticación de las personas. Las tarjetas inteligentes brindan efectividad, no solo para control de acceso físico sino también para nuevas aplicaciones y procesos que pueden beneficiar a la organización como un todo.

Derivado de la situación anteriormente expuesta se encuentra el siguiente **problema científico**: ¿cómo aumentar la seguridad de un sistema de control de acceso utilizando tarjetas inteligentes?

Como **objeto de estudio** se tiene: las tecnologías para desarrollar sistemas de control de acceso. El **campo de acción** se enmarca en: las tecnologías para desarrollar sistemas de control de acceso utilizando tarjetas inteligentes.

Para dar solución al problema existente se ha tomado como **objetivo general**: desarrollar el análisis, diseño e implementación de componentes `applet` y `middleware` para gestionar el acceso a instalaciones mediante la utilización de tarjetas inteligentes.

Idea a Defender:

Con la implementación de un `applet` y su `middleware` correspondiente, para su utilización en sistemas de control de acceso que utilicen tarjetas inteligentes se incrementará la seguridad de las instalaciones que utilicen esta tecnología.

Para dar respuesta a la interrogante presentada en este trabajo y con los objetivos trazados se plantea el cumplimiento de las siguientes **tareas de la investigación**:

- Caracterizar las tecnologías relacionadas con el desarrollo de sistemas de control de acceso a instalaciones.
- Caracterizar las tecnologías y estándares relacionados con tarjetas inteligentes.
- Realizar un estudio de la autenticación mediante patrones biométricos.
- Realizar un estudio para la selección de la metodología y herramientas ideales para el desarrollo de la solución.
- Elaborar la documentación y diagramas de ingeniería de software necesarios para el análisis y diseño de la solución.
- Desarrollar un `applet` haciendo uso de estándares y normas establecidas para esta tecnología.
- Desarrollar un `middleware` para establecer una comunicación con las funcionalidades del `applet` embebido en la tarjeta.
- Realizar las pruebas de calidad a la solución desarrollada.
- Realizar un prototipo de servicio que utilice esta solución.

La investigación estará sustentada en los siguientes **métodos científicos**:

Teórico: la utilización del método **Analítico-Sintético** posibilitará la revisión de fuentes bibliográficas y la fundamentación de los elementos más importantes que tienen relación con el objeto de estudio, el cual será de mucha importancia para el estado del arte.

También el método **Histórico-Lógico** brindará las maneras y las líneas de actuación generales que se deben seguir para acceder a la esencia del problema, además la compresión lógica del objeto de estudio haciendo un análisis riguroso de sus antecedentes y el proceso evolutivo por el cual han transitados todas las tecnologías relacionadas con las tarjetas inteligentes y los sistemas de controles de acceso.

Además el método **Hipotético-Deductivo** será de mucha ayuda partiendo de datos concretos para sacar conclusiones generales y viceversa, además de ser importante para la investigación.

Se usará el método de **Modelación** en la representación de las características fundamentales de la solución, a través de los diagramas y modelos que se realizarán a lo largo de la misma.

Empírico:

De los métodos empíricos se utilizará la Entrevista que sirve para la recopilación de información, ideas y opiniones acerca de la investigación contribuyendo con el desarrollo de la misma.

El trabajo se encuentra estructurado de la siguiente forma:

Capítulo 1 Fundamentación teórica: este capítulo contiene una base teórica para entender el problema planteado, en él se describen los conceptos fundamentales relacionados con tarjetas inteligentes, las tecnologías relacionadas con los sistemas de control de acceso y las metodologías a utilizar en el desarrollo del software.

Capítulo 2 Propuesta de solución: se presentan las fases de Exploración y Planificación definidas por la metodología XP (Extreme Programming, por sus siglas en inglés) para dar solución al problema científico. Se identifican las historias de usuarios y los requerimientos no funcionales, se realiza el plan de iteraciones y plan de entrega.

Capítulo 3 Implementación y prueba: en este capítulo se da cumplimiento a los planes trazados a través de las fases: iteraciones a primera liberación y Producción, se codifica la solución diseñada y finalmente se realizan las pruebas de aceptación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción

Las tarjetas inteligentes surgen en la década de los años 70, precisamente en Europa, alcanzando su mayor auge en los noventa, con la introducción de las tarjetas SIM [1] utilizadas en la telefonía móvil GSM[2]. El transcurso de los años ha propiciado que dichas tarjetas sean parte de los adelantos tecnológicos, por lo que su uso se hace cada día más necesario, dentro de las ramas de la informática que las utilizan están: la telefonía celular, el comercio electrónico y los bancos, donde las tarjetas inteligentes dan mayor seguridad que las magnéticas; además los sistemas de control de acceso mediante tarjetas inteligentes en algunos países cumplen función de documento de identificación debido a la seguridad tanto física como lógica que brindan.[10]

En el presente capítulo se hará un estudio de los principales sistemas de control de acceso existentes, principalmente los basados en las tarjetas inteligentes y como resultado se propone el desarrollo de una aplicación que resuelva los problemas que presentan continuamente estos sistemas. De dichas tarjetas se presenta un grupo de herramientas para la programación, estándares internacionales, características, aplicaciones y algunos conceptos con los que se relacionan. Se analizarán las soluciones factibles existentes y en base a su estudio se propondrá el desarrollo de una aplicación que resuelva el problema.

1.2 Conceptos asociados al dominio del problema

A continuación se relacionan algunos conceptos fundamentales para el entendimiento del objeto de estudio.

[1] Acrónimo de Subscriber Identity Module, Módulo de Identificación del Suscriptor en español.
[2] Acrónimo de Global System for Mobile Communications, Sistema Global para las Comunicaciones Móviles, en español.

1.2.1 Middleware

El middleware puede ser considerado como un software de conectividad que permite la interconexión entre diferentes aplicaciones, el acceso a las funcionalidades y a los datos de estas, desde y a través de otros sistemas independientemente de la plataforma. El middleware asociado al applet, que corre sobre el sistema operativo del chip embebido en la tarjeta inteligente funciona como una capa de software intermediario, que se intercala entre las aplicaciones del usuario y la tarjeta.[12]

1.2.2 APDU

Es una unidad de datos de protocolo de aplicación (Application Protocol Data Unit), usado por el middleware para enviar información al applet y APDU de Respuesta (R-APDU), usado por el applet para responder el comando enviado por el middleware. Su estructura es definida por el estándar ISO 7816.[13]

1.2.3 Applet

Los applets, en el contexto de las tarjetas inteligentes, son aplicaciones implementadas utilizando la tecnología JavaCard y ejecutadas dentro de estas. Un applet gestiona la información de las tarjetas inteligentes, además se ejecuta en el contexto del JCRE (JavaCard Runtime–Environment, por sus siglas en inglés), es el encargado de ejecutar las operaciones que maneja, mediante los comandos APDU que le son enviados, retornando los resultados mediante APDU de respuestas.[14]

1.2.4 Identificación

La identificación es el acto de reconocer si una persona es quien dice ser, a la hora de presentar datos para probar quién es. Los sistemas de control de acceso pueden tener diferentes tipos de identificación, en este caso una tarjeta inteligente posee todos los datos necesarios para reconocer al portador.[15]

1.2.5 Autenticación

La autenticación es el proceso de detectar y comprobar la identidad de una entidad de seguridad, mediante el examen de las credenciales del usuario y la validación de las mismas consultando a una autoridad determinada. Es el acto de establecimiento o confirmación de algo (o alguien) como auténtico. La autenticación de un objeto puede significar la confirmación de su

procedencia, mientras que la autenticación de una persona consiste en verificar su identidad. La misma depende de uno o varios factores.[16]

1.2.6 Autorización

La autorización es la acción y efecto de autorizar (reconocer la facultad o el derecho de una persona para hacer algo).[17] En el campo de la informática, la autorización es la parte de un sistema operativo que protege los recursos del sistema, de modo tal que sólo puedan ser utilizados por los usuarios que cuentan con permiso para eso. La autorización, por lo tanto, es una especie de permiso. Consiste en dar consentimiento para que otros hagan o dejen de hacer algo.[18]

1.3 Aplicaciones y características de las tarjetas inteligentes

1.3.1 Estructura de una tarjeta inteligente

Una tarjeta inteligente es un dispositivo similar a una tarjeta de crédito, el cual almacena y procesa información mediante un circuito de silicio embebido en el plástico de la tarjeta de acuerdo con el estándar ISO / IEC 7810 y ISO / IEC 7816.[19]

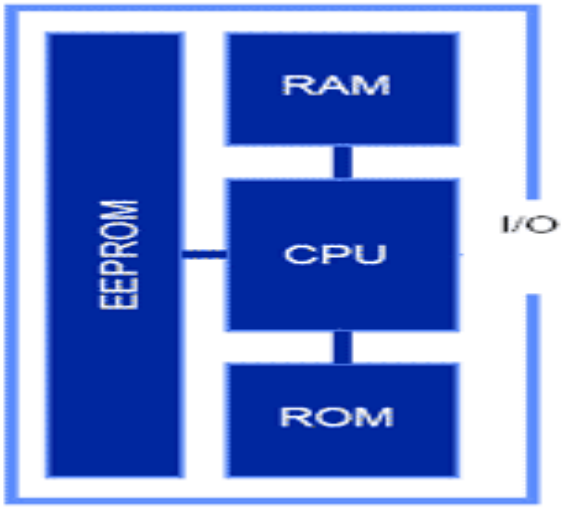

Figura 1: Estructura interna de una tarjeta inteligente.[20]

Las tarjetas inteligentes contienen un microprocesador de 8 `Bytes` con su **CPU** (Central Processing Unit, ¨Unidad Central de procesamiento¨), su **RAM** (Random-access memory, en español ¨Memoria de Acceso Aleatorio) y su **ROM** (Read Only Memory, en español ¨Memoria

de Solo Lectura¨), su forma de almacenamiento puede ser **EPROM** (Erasable Programmable Read Only Memory, en español ¨Memoria Programable y Borrable de Solo Lectura¨) o **EEPROM** (Electrically-Erasable Programmable Read Only, en español ¨Memoria Programable y Borrable Eléctricamente de Solo Lectura¨), el programa ROM consta de un sistema operativo que maneja la asignación de almacenamiento de la memoria, la protección de accesos y maneja las comunicaciones. El sendero interno de comunicación entre los elementos (BUS) es totalmente inaccesible desde afuera del `chip`, por ello la única manera de comunicación está bajo control del sistema operativo y no hay manera de poder introducir comandos falsos o requerimientos inválidos, que puedan sorprender las políticas de seguridad.[19]

Internamente, el `chip` de una tarjeta inteligente con microprocesador se compone de:

CPU: el procesador de la tarjeta; suele ser de 8 bits, a 5 MHz y 5 voltios. Pueden tener opcionalmente módulos hardware para operaciones criptográficas.

ROM: memoria interna, (normalmente entre 12 y 30 KB), en la que se establece el sistema operativo de la tarjeta, las rutinas del protocolo de comunicaciones y los algoritmos de seguridad de alto nivel. Esta memoria, como su nombre indica, no se puede reescribir y se inicializa durante el proceso de fabricación.

EEPROM: memoria de almacenamiento, (equivalente al disco duro en un ordenador personal), en el que está grabado el sistema de ficheros, los datos usados por las aplicaciones, claves de seguridad y las propias aplicaciones que se ejecutan en la tarjeta. El acceso a esta memoria está protegido a distintos niveles por el sistema operativo de la tarjeta.

RAM: memoria volátil de trabajo del procesador.

La tarjeta sólo reacciona a los requerimientos de datos externos, nunca inicia por sí sola una comunicación. Todo el protocolo de comunicaciones, dimensiones, resistencia, etc., está claramente establecido en el estándar ISO-7816. Dentro de la categoría de tarjetas inteligentes, con microprocesador se encuentran las llamadas `JavaCards` o `JavaSmartCards`. Una `JavaCard` es una `SmartCard` capaz de ejecutar programas desarrollados en Java. Concluyendo, una `JavaSmartCard` es una tarjeta con microprocesador que puede ejecutar programas (llamados `applets`) escritos en un subconjunto del lenguaje `Java`.[19]

1.3.2 Algunas clasificaciones de las tarjetas inteligentes.[21]

- **Según la interfaz de comunicación.**

De acuerdo con la interfaz de comunicación las tarjetas inteligentes pueden ser clasificadas en:

- **Tarjeta de contacto:** requieren ser insertadas en un terminal con lector inteligente para que pueda ser leída. Las características físicas de esta tarjeta están especificadas en el estándar ISO 7816 donde existe la estructura de un conjunto de 8 pines a través de los cuales se establece la comunicación con el lector.
- **Tarjeta sin contacto:** utilizan diferentes protocolos de transmisión en capa lógica y física, no utiliza contacto galvánico sino que establecen comunicación a través de campos electromagnéticos usando una antena, lo que permite que se use desde media distancia sin necesidad de ser introducida en un lector. El estándar de comunicación de tarjetas inteligentes sin contacto es el ISO/IEC 14443, el cual incluye las especificaciones de las características físicas, la energía de radiofrecuencia, la señal de interfaz, protocolos de inicialización, de anticolisión y de transmisión. Una de las ventajas que esta tarjeta tiene, es que al no existir contactos externos, esta es más resistente a los elementos tales como la mugre y no es dañada por el rozamiento.
- **Tarjeta híbrida y dual**: estas tarjetas contienen las 2 topologías explicadas anteriormente. Las híbridas contienen 2 `chips` totalmente independientes, uno para la comunicación por contacto y otro para la comunicación sin contacto. En el caso de la de interfaz dual, esta contiene un único `chip` con ambas interfaces.

- **Según la estructura de su sistema operativo.**

De acuerdo a la estructura de su sistema operativo se pueden clasificar en:

- **Tarjetas de memoria**: disponen de un sistema operativo limitado a una serie de comandos básicos de lectura y escritura de las distintas secciones de memoria y la protección de la información está condicionada a la presentación de un código secreto.
- **Tarjetas basadas en ficheros**: estas tarjetas disponen del equivalente a un sistema de ficheros MS-DOS con dos niveles de jerarquía. Hay directorios y ficheros. Tienen un sistema operativo con un conjunto de comandos que le ofrecen las operaciones básicas para el acceso a los datos y la protección de la información.
- **Tarjetas Java**: son tarjetas capaces de ejecutar mini-aplicaciones Java. Su sistema operativo es una pequeña JVM (en español, ¨Máquina Virtual de Java¨) y en ellas se

pueden cargar dinámicamente aplicaciones desarrolladas específicamente para este entorno.

1.3.3 Protocolo de comunicación en tarjetas inteligentes.

Unidad de datos de protocolo de aplicación (Application Protocol Data Unit), es la unidad de comunicación entre un lector y una tarjeta. Su estructura está definida en el estándar ISO 7816, existiendo dos tipos de categorías de APDU, APDU Command (Comando APDU) y APDU Response (APDU Respuesta).

Los comandos APDU son utilizados para la comunicación entre el `applet` y el `middleware`, desde una tarjeta nunca se transmite información sin que antes se haya producido una petición externa por el `middleware` a través del dispositivo lector. Para identificar el APDU que recibe la tarjeta y el que envía como respuesta se denomina Comando APDU (C-APDU), usado por el `middleware` para enviar información al `applet` y APDU de Respuesta (R-APDU), usado por el `applet` para responder el comando enviado por el `middleware`.[22]

Una vez que la tarjeta recibe el APDU, lo procesa y devuelve a la aplicación externa una notificación que indica si el proceso concluyó, o si ocurrió un error al procesar el APDU recibido. La estructura general del C-APDU lo conforman siete elementos, sin embargo no siempre se envían los siete elementos, pueden ser enviados los primeros cuatro ya que los datos (Data Field) son opcionales, debido a que no siempre la tarjeta necesita información del exterior en cada APDU que recibe. La siguiente figura muestra la estructura de un C-APDU, como lo define el estándar ISO/IEC 7816-4.[13]

APDU Command (C-APDU: este comando es usado por el lector para enviar información a la tarjeta).

Encabezado (obligatorio)				Cuerpo (opcional)		
CLA	INS	P1	P2	Lc	Data Field	Le

Figura 2: Estructura Comando APDU.[13]

CLA: clase de instrucción. Indica la estructura y el formato.

INS: código de instrucción. Especifica la instrucción del formato.

P1, P2: parámetros de instrucción. Proveen más información sobre la instrucción.

LC: número de `bytes` en el Data Field del APDU.

Data Field: secuencia de `bytes` con información.

LE: cantidad máxima de `bytes` esperados como respuesta.

APDU Response (R-APDU: este comando es usado por la tarjeta para responder al comando enviado por el lector.)

Cuerpo (opcional)	Código respuesta (obligatorio)	
Data Field	SW1	SW1

Figura 3: Estructura APDU Respuesta.[13]

Data Field: secuencia de `bytes` con información.

SW1, SW2: `Status Word` (palabra de estado): denotan el estado del procesamiento del comando en la tarjeta.

1.3.4 Aplicaciones de las tarjetas inteligentes

En la actualidad las tarjetas inteligentes tienen un gran número de aplicaciones, entre las que se puede mencionar el control de acceso; este tipo de aplicación suele estar ligada a puertas automatizadas que permiten o impiden el paso físico de una persona a un área determinada. También son muy utilizadas como monederos electrónicos, a finales del siglo pasado se registraban ya millones de usuarios que las usaban con este fin. Estas disponen normalmente de un fichero protegido que almacena un contador de saldo y comandos que disminuyen y aumentan el mismo (esto último sólo con claves de seguridad especiales). Con esta aplicación, el `chip` de la tarjeta inteligente puede ser ¨cargado¨ con dinero (en terminales autorizados que dispongan de las claves de seguridad). Este dinero virtual puede ser utilizado en parquímetros, máquinas expendedoras u otros mercados. El monedero electrónico extiende el uso de las tarjetas a acciones tan simples como el pago de televisión, telefonía móvil, transporte público, acceso a sitios `web`, entre otros.

Otra importante aplicación en la actualidad es la firma digital de documentos, o sea la posibilidad de almacenar un certificado digital dentro de la tarjeta y firmar con él documentos electrónicos sin que en ningún momento el certificado (y más concretamente su clave privada) salgan del almacenamiento seguro en el que están confinados. El uso de la tarjeta anula la

necesidad de tener que compartir una inmensa base de datos y tener que hacer réplicas periódicas.

Esta misma facilidad es la que se aprovecha al utilizar las tarjetas inteligentes en clínicas y sistemas nacionales de salud donde se está implementando un sistema de identificación de pacientes y control de los datos del historial clínico o información relativa a enfermedades crónicas o alérgicas dentro de la tarjeta personal.[23]

Otro ejemplo de las aplicaciones de las tarjetas inteligentes es el DNI español, el cual utiliza una tarjeta de policarbonato con `chip` integrado y se utiliza tanto para la identificación presencial del portador, como para la identificación electrónica, mediante los certificados contenidos en la memoria no volátil del `chip`. También se utiliza en el sector privado para: acceder a sitios `web`, firma de contratos, verificación de autenticidad de documentos y en el sector público para la declaración anual de impuestos, interacción con la Administración Pública para la obtención de formularios, servicios en línea, y registros criminales. Como se han descrito las aplicaciones de las tarjetas inteligentes son muy variadas.

El control de acceso mediante tarjetas inteligentes es una aplicación muy usada por grandes instalaciones, en las cuales se hace muy difícil la supervisión de la entrada/salida de las personas, lo cual representa un peligro para la seguridad de dichas instalaciones.

1.3.5 Seguridad asociada a las tarjetas inteligentes

Las tarjetas inteligentes son ampliamente reconocidas como unas de las formas más seguras y fiables de la identificación electrónica. Poseen un `chip` a prueba de falsificación que no puede ser duplicado, posee capacidad de detección de ataques por rayos X y luz ultravioleta, voltajes inusuales y cambios de frecuencia de reloj. Sus medidas de protección abarcan en toda la extensión desde la elaboración y los elementos empleados en el material de la misma hasta las características usadas en su impresión gráfica dificultando su reproducción y alteración. Su sistema operativo posee control del acceso a memoria, protección de datos y ficheros.[24]

La seguridad de las tarjetas inteligentes se aplica a múltiples niveles y con distintos mecanismos, es una de las propiedades más importantes que brinda este tipo de tarjetas. Cada fichero lleva asociadas unas condiciones de acceso que deben ser satisfechas antes de

ejecutar un comando sobre ese fichero. A los mismos se les aplican diferentes mecanismos de seguridad durante la fabricación, estos son:

- **Ficheros de acceso libre.**
- **Ficheros protegidos por claves:** claves que se definen para proteger la escritura de ficheros. Por lo que cualquier aplicación que intente ejecutar comandos sobre ficheros protegidos tendrá que autenticarse con la clave correcta.
- **Ficheros protegidos por PIN**: al usuario introducir el PIN, el programa se lo pasa a la operación que va a abrir el fichero en cuestión y el sistema valida que el PIN sea correcto para dar acceso al fichero.

1.3.6 Proceso de gestión de control de Acceso mediante tarjetas inteligentes

El proceso de control de acceso propuesto, comienza cuando un usuario introduce la tarjeta en el lector. Luego estos son procesados, si el PIN incorporado al sistema es correcto, el lector autentica los datos. La validez es determinada desde dentro de la misma tarjeta inteligente al comparar el PIN introducido por el portador, con el almacenado en la tarjeta, posteriormente el sistema muestra un mensaje informando que el PIN es correcto, en caso contrario el mensaje sería para informar el error cometido. Luego se realizan otras acciones como, lectura y escritura de los datos del portador de la tarjeta, además de permitir un cambio de PIN, si el usuario lo solicita. Para mitigar los riesgos contra accesos no autorizados o ataques deliberados, se implementó la configuración de canal seguro, además de utilizar autenticación por PIN.

1.3.7 Almacenamiento seguro de Datos en la tarjeta

A la hora de realizar un cambio en la estructura de datos existente en la tarjeta inteligente, existen dos necesidades principales: el almacenamiento de la información del usuario, cambio de dicha información. En cuanto al almacenamiento de la información de la tarjeta, hay que tener en cuenta el tamaño de la misma. La modificación de la información solo se realiza con respecto a las condiciones de acceso.

La tecnología de tarjetas inteligentes en conjunto con el chip, el resultado es un dispositivo que puede respaldar una amplia variedad de aplicaciones seguras. Las únicas restricciones técnicas son el tamaño físico del chip y la cantidad de memoria disponible. Debido a la alta

seguridad de las tarjetas, las mismas poseen múltiples aplicaciones. El acceso a los registros es típicamente protegido con algún tipo de lógica de control de acceso, en este caso un PIN.

Los modos de acceso sobre la información son: solo lectura de la información, solo escritura y lectura / escritura de la misma.

1.3.8 Necesidad de las tarjetas inteligentes en la autenticación.

En infinidad de instalaciones a nivel mundial la autenticación del personal es intrínseca a la entrada y salida del mismo, por lo que son designados informáticos a esta tarea para que solucionen los problemas que se presentan con respecto a este tema, para que implementen sistemas de control de acceso. Existen diferentes tipos de estos sistemas, dentro de ellos están los biométricos, que utilizan las huellas dactilares, lectura de iris, entre otros, pero estos presentan debilidades difíciles de solucionar y que conllevarían a un error en el acceso, lo cual podría representar un desastre para la seguridad de dichas instalaciones.

Las tarjetas inteligentes se pueden utilizar en diferentes escenarios, tales como autenticación de clientes, autenticación de acceso remoto y firmas de correo electrónico. Cuando se tiene una tarjeta (algo que se posee) y un PIN privado (algo que se conoce), es menos probable que una persona pueda asumir la identidad de otra. La única manera en que esto sucedería es al tener acceso tanto al dispositivo físico como al PIN. Este tipo de autenticación y las soluciones de los proveedores para administrar la complejidad de estos sistemas, logran que éste sea un proyecto razonable para resolver el desafío de la autenticación.[9]

1.4 Estándares relacionados con tarjetas inteligentes

1.4.1 Global Platform

`Global Platform` es una organización independiente enfocada a gestionar una infraestructura estandarizada para el desarrollo y despliegue de tarjetas inteligentes. Proporciona un conjunto de especificaciones universalmente reconocidas e implementadas, junto con configuraciones de mercado y aplicación de esas especificaciones y documentos de apoyo. Cubriendo toda la infraestructura de tarjetas inteligentes (las tarjetas, dispositivos y sistemas), estos documentos técnicos ofrecen una plataforma tecnológica dinámica y completa para el desarrollo de programas de tarjetas inteligentes, para poder establecer una conexión segura con la misma y administrar sus aplicaciones.

Las tarjetas, dispositivos y sistemas Global Platform, son interoperables, independientemente de la tecnología del proveedor y la flexibilidad de su infraestructura técnica, garantizan que se pueda responder a las necesidades básicas en el instante del despliegue inicial. Ofreciendo a los emisores la seguridad de que la infraestructura que han elegido será capaz de adaptarse y crecer a medida que cambian las condiciones de negocios.[25]

1.4.2 Estándar ISO/IEC- 7816

La tarjeta inteligente más básica cumple los estándares de la serie ISO 7816. El objetivo de estos es lograr la interoperabilidad entre distintos fabricantes de tarjetas y lectores de las mismas, en lo que respecta a características físicas, comunicación de datos y seguridad. Estos estándares son basados en los ISO 7810 e ISO 7811, los cuales definen características físicas de tarjetas de identificación. Las características de comunicación de las tarjetas sin contacto son definidas en estándares como el ISO/IEC 14443.[26]

Descripción de algunas partes del estándar ISO 7816:[27]

- 7816-1: Características físicas.
- 7816-2: Dimensiones y ubicaciones de los contactos.
- 7816-3: Señales electrónicas y protocolo de transmisión.
- 7816-4: Comandos de intercambio inter-industriales.
- 7816-5: Sistema de numeración y procedimiento de registración.
- 7816-6: Elementos de datos inter-industriales.
- 7816-7: Comandos inter-industriales y consultas estructuradas para una tarjeta.
- 7816-8: Comandos inter-industriales relacionados con seguridad.
- 7816-9: Comandos adicionales inter-industriales y atributos de seguridad.
- 7816-10: Señales electrónicas y respuesta al reset para una tarjeta inteligente síncrona.

1.4.3 Estándar PC/SC

PC/SC [3] es un conjunto de especificaciones para la integración de tarjetas inteligentes en ordenadores personales. En particular se define un API[4], que permite a los desarrolladores

[3] En inglés Personal Computer/ Smart Card.
[4] Aplication Programming Interface, por sus siglas en inglés, en español Interfaz de Programación de Aplicaciones,

trabajar de forma uniforme con lectores de tarjetas de distintos fabricantes (que cumplan con la especificación).

El API de PC/SC está incorporado en sistemas Microsoft Windows 200x/XP y Microsoft Windows NT/9x. También hay una implementación libre, de código abierto, llamada PC/SC Lite (proyecto MUSCLE) para sistemas operativos GNU Linux.

La especificación se divide en 10 partes que contienen los requisitos detallados, interoperabilidad de dispositivos compatibles, información de diseño, interfaces de programación y otras.[28]

- Parte 1. Introducción y visión general de la arquitectura.
- Parte 2. Requisitos de interoperabilidad para las tarjetas y los lectores.
- Parte 3. Requisitos de interoperabilidad para los lectores conectados.
- Parte 4. Consideraciones de diseño e información de referencia de los lectores.
- Parte 5. Definición de la interfaz del Resource Manager.
- Parte 6. Definición de la interfaz del Service Provider.
- Parte 7. Consideraciones de diseño para el desarrollo de aplicaciones.
- Parte 8. Recomendación para la implementación de servicios de seguridad y privacidad con tarjetas inteligentes.
- Parte 9. Lectores con capacidades extendidas.
- Parte 10. Lectores con capacidades de entrada de PIN de seguridad.

1.5 Sistema de control de acceso

Los sistemas de control de acceso se hacen cada día más necesarios debido a la existencia de información a proteger y bienes, tanto digitales, como en formato duro; el auge tecnológico existente se convierte en una ventaja para esta necesidad, posibilitando la implementación de nuevos y sofisticados sistemas que resuelven problemas y mejoran los ya existentes. Estos sistemas son de gran interés, sobre todo cuando los recursos a proteger se encuentran en un entorno con grandes comunidades de usuarios y se necesita un límite de acceso a dichos recursos.[2]

1.5.1 Sistema para el reconocimiento de placas de matrícula.

Existen sistemas para el reconocimiento de placas de matrícula que capturan, procesan, interpretan y graban imágenes del vehículo para su uso en cualquier aplicación. Estos son ampliamente usados en grandes edificaciones que poseen una concurrente entrada y salida de automóviles, además en casas, embajadas y zonas militares o políticas donde es necesario un control exhaustivo del flujo de personas.

Método para reconocer placas de matrícula empleando una cámara inteligente con un procesador y una memoria, comprendiendo las etapas del método: capturar mediante la cámara una imagen que incluye una placa de matrícula; detectar una región en la que se sitúa la placa de matrícula realizando una localización aproximada en la imagen; detectar las condiciones de orientación, posición e iluminación de la imagen para obtener una imagen de referencia de la placa de matrícula; realizar una localización precisa y una operación de nuevo muestreo para obtener una representación más precisa de resolución vertical de la imagen de referencia de la placa de matrícula; segmentar los caracteres representados en la imagen de referencia, empleando una proyección a lo largo de un eje horizontal de la imagen de referencia para identificar la posición de los caracteres; clasificar los caracteres basándose en un clasificador estadístico para obtener una puntuación de seguridad de la probabilidad de identificar correctamente cada carácter y realizar de manera recursiva la etapa del método de segmentación de caracteres y las siguientes etapas del método hasta que cada puntuación de seguridad supera un valor umbral para reconocer los caracteres; en el que después de la etapa de segmentación de caracteres y antes de la etapa de clasificación de caracteres se realiza la etapa de comparar cada carácter en la imagen de la placa de matrícula con ejemplos de imágenes con diferentes iluminaciones para tener en cuenta los efectos de iluminación sobre la imagen.[29]

1.5.2 Lectores de Control de Acceso

Un lector, en este caso, es un dispositivo que se puede tener diferentes características, aunque en el mercado se encuentra principalmente en tres formas: integrado en el teclado, en un dispositivo USB que se conecta al ordenador o como tarjeta PCMCIA[5]. Existen una variedad de fabricantes distintos como Bit4id, C3PO, Gemalto o Kalysis, por citar algunos.

[5] Acrónico de Personal Computer Memory Card International Association.

El dispositivo USB [6] suele ser el más barato y su tamaño reducido permite ser trasladado con facilidad si se tiene pensado usarlo en más de un lugar. El mismo consta de una entrada USB y un pequeño adaptador que lee la información que contiene el `chip` dorado que caracteriza a las tarjetas electrónicas.

Los lectores que se integran en el teclado son más caros, porque para adquirirlos hay que comprar un teclado que ya lo lleve incorporado, con lo que se suma el coste del mismo. Su ventaja se aprecia si se emplean las tarjetas inteligentes con asiduidad, porque siempre están a mano del usuario y no hay que preocuparse de enchufar o desenchufar ningún USB. Sin embargo, ofrecen menos portabilidad.[5]

Los lectores basados en tarjetas PCMCIA son dirigidos a los usuarios con ordenadores portátiles. Se pueden usar con más comodidad, sobre todo en movimiento, que los dispositivos USB porque se integran dentro del cuerpo del portátil.

En principio, cualquiera de los lectores de tarjetas se puede utilizar en los sistemas operativos más habituales (Windows, Mac OS X y GNU/Linux). Estos equipos suelen traer de fábrica un disco con los controladores necesarios para que funcione el equipo en el ordenador. Cada tarjeta inteligente o certificado de identidad necesita que los lectores cumplan unas condiciones mínimas que señalan sus promotores. Esta información se debe comparar con las especificaciones técnicas de los lectores, aunque la gran mayoría cumple con todos los requisitos.

1.5.3 Sistemas biométricos

Se entiende por sistema biométrico, un sistema automatizado que realiza labores de biometría. Es decir, un sistema que fundamenta sus decisiones de reconocimiento mediante una característica personal que puede ser reconocida o verificada de manera automatizada.

En la actualidad existen sistemas biométricos que basan su acción en el reconocimiento de diversas características. Las siguientes técnicas biométricas son las más conocidas y están basadas diferentes indicadores biométricos:[30]

(Ver Anexo 1)

[6] Acrónico de Universal Serial Bus.

- Reconocimiento de rostro: una modalidad biométrica que utiliza una imagen de la estructura física de la cara visible de una persona para fines de reconocimiento.
- Termograma del rostro: cámaras infrarrojas reconocen la distribución de calor en la imagen.
- Reconocimiento de huellas dactilares: utiliza la estructura física de las huellas dactilares de un individuo.
- Reconocimiento de la geometría de la mano: utiliza la estructura física de la mano de un individuo.
- Reconocimiento de las venas de las manos: las venas poseen infinitas características que las hacen únicas.
- Reconocimiento del iris: se basa en capturar la imagen del iris y obtener de esta un patrón denominado código del iris.
- Patrones de la retina: el escáner de retina ilumina, a través de la pupila, una región de la retina con luz infrarroja y almacena la información del contraste de los patrones vasculares reflejados.
- Voz: en el reconocimiento de voz se comparan características tales como calidad, duración intensidad dinámica.
- Firma: el proceso de la firma se origina en unas propiedades intrínsecas del sistema neuromuscular del ser humano, que produce los movimientos rápidos.

1.5.4 Uso de las tarjetas inteligentes en los sistemas de control de acceso.

Dentro de los sistemas de control de acceso existentes, están los basados en dispositivos inteligentes de identificación, hoy en día existe una tendencia creciente al uso de los mismos, entre los que se destacan las tarjetas inteligentes, dichas tarjetas proporcionan un modelo totalmente renovador en cuanto a sistemas de control de acceso se trata, debido principalmente a que fortalece la seguridad de estos sistemas. La necesidad de una credencial segura para el acceso a instalaciones, es uno de los requisitos más importantes en cuanto al uso de las tarjetas inteligentes en este campo.

Los sistemas de control de acceso físico basados en tarjetas inteligentes son una herramienta de seguridad poderosa y eficiente para proteger los bienes de una instalación. Aminorando los principales riesgos ya que protegen la información que almacenan en sus memorias de posibles accesos no autorizados y poseen una mayor resistencia al deterioro de la información

almacenada. Poseen un alto nivel de seguridad, por lo que son confiables, su uso es fácil y mantienen los datos del titular bajo total privacidad.[8]

1.5.5 Análisis de otras soluciones

Los sistemas de control de acceso a recursos informáticos, son de enorme interés sobre todo cuando dichos recursos generalmente están limitados y el entorno es de grandes comunidades de usuarios. Por lo que se hace necesario una investigación profunda de distintos sistemas de control de acceso actuales que utilicen las tarjetas inteligentes.

1.5.5.1 Sistema de control de acceso basado en los lectores de control de acceso V-Smart iCLASS y tarjetas inteligentes del Aeropuerto internacional de Ciudad de México (AICM).

El AICM solicitó ayuda a HID Global, fabricante destacado en la industria del control de acceso, para dar solución a sus necesidades. Basándose en la evaluación de la amplitud de la actualización y en su experiencia previa con los productos de HID Global, el AICM se sentía seguro con la adquisición de hardware de control de acceso. El AICM configuró un nuevo sistema de control de acceso basado en los lectores de control de acceso V-Smart iCLASS y tarjetas inteligentes sin contacto de 16 kbits (2 kbytes).

El equipamiento, suministrado por Bioscrypt e HID, requiere autenticación biométrica además de la verificación de la identidad de los portadores de las tarjetas, para obtener acceso a las zonas restringidas. Los lectores de huellas dactilares incluyen la tecnología de tarjetas inteligentes sin contacto de lectura/escritura HID iCLASS® de 13,56 MHz para administrar el acceso a sitios restringidos dentro del aeropuerto como, por ejemplo, las salas VIP [7]y las zonas de operaciones. Si alguien desea pasar por una puerta controlada, primero debe identificarse presentando su tarjeta de control de acceso. Una vez que la tarjeta se lee y verifica correctamente, el portador coloca el dedo en el lector biométrico para demostrar qué es realmente la persona para la que se emitió. De esta forma, el acceso es prácticamente imposible con una tarjeta que no pertenezca al propietario.[31]

1.5.5.2 Registro de asistencia basado en tarjeta inteligente Janus.

Es un terminal que permite de forma sencilla e intuitiva registrar la presencia en aulas, centros de trabajo para usuarios de cualquier índole basándose simplemente en una tarjeta inteligente

[7] Acrónimo de Very Important Person.

sin contactos. Es un sistema de captura de marcajes mediante tarjeta inteligente y terminales autónomos empotrados en pared. Es compatible con cualquier tipo de tarjeta que cumpla la norma ISO 14443. En la tarjeta inteligente se incorpora información personal del propietario siendo esta información, la que utiliza para identificarse ante el sistema. Estará protegida por un PIN personal que impide su uso fraudulento en caso de pérdida. Así mismo, existe la posibilidad de proteger el dispositivo con un PIN que únicamente conozca el profesor, así como de seleccionar la asignatura para la cual se registra la asistencia de entre las que se imparten en esa aula.

El escenario anterior a Janus, está protagonizado por actas en papel, en muchos de los casos incompletas, con datos falseados, o inexistentes. Los centros educativos en general, presentan un bajo porcentaje de éxito en la auditoría de asistencia que se caracteriza por procesos rudimentarios, tediosos y poco fiables.[32]

Este sistema aporta a todas las partes implicadas una serie de ventajas:

Desde el punto de vista de la universidad:

- Un nivel más profesional de control y auditoría en cuanto a la asistencia de alumnos y profesores.
- Mayor trazabilidad ocasionada por la obtención de marcajes en tiempo real.
- Aumento pronunciado de la fiabilidad y seguridad en los marcajes, repercutiendo favorablemente en el registro de incidencias ocasionadas por errores o falsificaciones en la identificación.
- Mayor eficiencia con menor coste humano en la gestión de RRHH.

Desde el punto de vista de los usuarios:

- Un sistema más cómodo, fiable y seguro que los anteriores.
- Eliminación de errores en la identificación.
- El dispositivo en el que se basa el sistema Janus, tiene una apariencia sencilla y amigable con pantalla táctil y menús simples que facilitan enormemente el proceso de marcaje:

Las características básicas del dispositivo son las siguientes:

- Diseño propiedad de Acotec Smartcard Solutions S.L. que dota a la solución de la posibilidad de adaptación a otros escenarios futuros (`chip` con contactos etc.)
- Pantalla táctil TFT de 5,7" y resolución de 320X240 pixeles.
- Lector de tarjetas compatible con norma ISO 14443 A y B.
- Conexión WIFI

1.5.5.3 Smart Acces Control, Sistema de Control de Acceso por Tarjeta Inteligente producido por ISV/Software Solutions.

Sistema de Control de Acceso por Tarjeta Inteligente de proximidad ISO 14443ª y biometría con huella dactilar y administración Web. Sistema de alto tráfico con huella dactilar utilizando tarjetas inteligentes sin contacto, bajo una plataforma de integración en software-hardware, una estructura de comunicación TCP/IP [8](LAN[9], MAN[10], WAN[11]) y base de datos MICROSOFT SQL/SERVER. La solución del sistema requerida es planteada, mediante una estructura altamente confiable y que soporte tráfico masivo sin retardos y problemas de encolamiento. Para ello se aplicó una estructura bajo TCP/IP con una base de datos altamente confiable y eficiente como es MICROSOFT SQL/SERVER, un software de registro, gestión, administración y control de tiempos, integrados en una plataforma robusta y un sistema de terminales biométricas (huella dactilar) con puerto Ethernet 10/100 base T con lectores de proximidad para tarjetas inteligentes sin contacto norma ISO 14443ª, aplicada a los ingresos y salidas del personal (funcionarios, proveedores, visitantes), así cada persona podrá acceder a las áreas o dependencias configuradas tanto en la base de datos como en la tarjeta inteligente. Las tarjetas inteligentes se pueden personalizar con fotografía, datos personales, logos y otros parámetros según necesidades del cliente.[33]

1.5.5.4 Comparación entre las soluciones analizadas y la desarrollada por CISED.

Luego de un estudio acerca de las soluciones anteriormente expuestas, queda demostrado que las mismas tienen como característica similar entre ellas y la solución propuesta por el presente trabajo de diploma, la utilización de la tecnología de tarjetas inteligentes para gestionar el control de acceso a instalaciones. Existen también una serie de diferencias, con respecto a los

[8] Protocolo de Control de Transmisión/Protocolo de Internet (en inglés Transmission Control Protocol/Internet Protocol).
[9] Una red de área local, red local o *LAN,* (en inglés Local Area Network).
[10] Una MAN Red de área metropolitana, (en inglés Metropolitan Area Network).
[11] Una WAN Red de Área Amplia, (en inglés Wide Area Network).

lectores de tarjeta, al tipo de tarjeta y por ende a las tecnologías. A continuación se exponen algunas semejanzas y diferencias.

- Lectores de tarjeta:
- AICM utiliza lectores V-Smart iCLASS.
- Janus utiliza lectores compatibles con la norma ISO 14443 A y B.
- Smart Acces Control utiliza lectores compatibles con la norma ISO 14443 A.
- La solución propuesta por el presente trabajo de diploma utiliza el lector SD 010 compatible con la norma ISO 7816.
- Tipos de tarjeta:
- AICM utiliza tarjetas sin contacto.
- Janus utiliza tarjetas sin contacto.
- Smart Acces Control utiliza sin contacto.
- La solución propuesta por el presente trabajo de diploma utiliza tarjetas con contacto.
- Formas de autenticación:
- AICM utiliza autenticación por huellas dactilares.
- Janus utiliza autenticación por PIN.
- Smart Acces Control utiliza autenticación por huellas dactilares.
- La solución propuesta por el presente trabajo de diploma utiliza autenticación por PIN.

1.6 Tecnologías relacionadas con el desarrollo

1.6.1 Metodologías de desarrollo

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software.

Todo producto software que requiera contar con una aplicación, ya sea por medio de un desarrollo, debe considerar algunos procedimientos, de acuerdo, a la normativa vigente. Todo documento que se genere durante el proceso, deberá quedar dentro del expediente del Proyecto de Desarrollo del Sistema de Información, así mismo la versión final de cada uno de los artefactos del proyecto debe apegarse a lo último implementado en cada caso, y por lo tanto será indispensable su actualización o refinamiento para la entrega final de la aplicación.

Capítulo 1: Fundamentación teórica.

La selección de una metodología de desarrollo de software adecuada, es un factor determinante en el éxito de un proyecto. Aunque no existe una metodología absoluta, algunas se ajustan mejor que otras, a las características y necesidades específicas de los proyectos de desarrollo.[34]

Dentro de las metodologías de desarrollo existen dos grandes grupos, las conocidas metodologías tradicionales y las metodologías ágiles. Las primeras enfatizan en el uso exhaustivo de documentación durante todo el ciclo de vida del proyecto y es recomendada para los proyectos con grandes equipos de desarrollo. Liderando este grupo se encuentra RUP[12], que es el resultado de varios años de desarrollo y del uso práctico, en el que se han unificado varias técnicas de desarrollo. Mientras que las ágiles dan mayor importancia a la capacidad de respuesta a los cambios, se enfatiza en la satisfacción del cliente y promueve el trabajo en equipo.

La metodología XP está diseñada para entregar el software que el cliente necesita, en el momento que lo necesita. Además promueve el uso de prácticas para aumentar la productividad del equipo de desarrollo y mejorar la adaptabilidad a los frecuentes cambios dentro del ciclo de vida del proyecto.[35]

Ventajas:

- Apropiado para entornos volátiles, equipos de desarrollo pequeños (de 2 a 10 desarrolladores) y proyectos de alto riesgo.
- Permite una mejor adaptabilidad a los cambios, que se traduce en una reducción de costos.
- Planificación a corto plazo y más transparente para los clientes, ya que conocen las fechas de entrega de funcionalidades vitales para su negocio.
- Permite tener retroalimentación continua de los usuarios a través de las entregas frecuentes.

Desventajas:

- A veces cuesta delimitar el alcance del proyecto con el cliente.

[12]Acrónimo de Rational Unified Process.

1.6.1.1 Fundamentación de XP como metodología a utilizar

Entre los elementos determinantes para su elección fueron:

- El tamaño del grupo de desarrollo, en este caso, de apenas dos personas.
- Cliente presente en el grupo de desarrollo.
- Necesidad de resultados tangibles a corto plazo.
- Imposibilidad, para un grupo de desarrollo pequeño, de asumir una metodología robusta, debido a la cantidad excesiva de roles y documentación generada en el ciclo de vida del proyecto.

Valores, prácticas y variables de XP

XP es una metodología de desarrollo ligera basada en una serie de valores, principios y una docena de prácticas que propician un aumento en la productividad a la hora de generar software.[36]

- **Valores que promueve**

En el ciclo de vida de un proyecto los cambios van a aparecer y a veces el equipo de desarrollo no está preparado para enfrentarlos, XP desarrolla los siguientes valores para garantizar el éxito de un proyecto de desarrollo de software:

- Comunicación.
- Coraje.
- Simplicidad.
- Retroalimentación.

- **Prácticas**

La mayoría de estas prácticas no son nuevas, sino que han sido reconocidas por la industria como las mejores prácticas durante años.

- Planificación Incremental.
- Pruebas.
- Programación en parejas.
- Refactorización.
- Diseño simple.
- Propiedad colectiva del código.

- Integración continua.
- Cliente en el equipo.
- Entregas pequeñas.
- Semanas de 40 horas.
- Estándares de codificación.
- Uso de Metáforas.

➢ **Variables**

XP define cuatro variables para proyectos de software: coste, tiempo, calidad y ámbito. Además de estas cuatro variables, Beck el creador de la metodología XP, propone que sólo tres puedan ser establecidas por las fuerzas externas (jefes de proyecto y clientes), mientras que el valor de la cuarta variable debe ser establecido por los programadores en función de las otras tres.

Artefactos esenciales en XP

- Historias del usuario.
- Tareas de ingeniería.
- Pruebas de aceptación.
- Pruebas unitarias y de integración.
- Plan de entrega.
- Código

Fases de XP

La metodología XP, como metodología ágil, enfatiza en el carácter interactivo e incremental del desarrollo, donde una iteración es un período de una a cuatro semanas, en el cual el cliente selecciona las funcionalidades que desea que se implementen en dicha iteración. La figura muestra las fases en las que se subdivide el ciclo de vida de un proyecto de software con XP. (Ver anexo 2)

➢ **Exploración**

En esta fase, los clientes plantean a grandes rasgos las Historias de Usuario (HU) que son de interés para la primera entrega del producto, además se confecciona la metáfora del sistema ayudando al equipo a entender las relaciones entre los principales componentes del sistema. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y

prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.

- **Planeamiento**

En esta fase el cliente establece la prioridad de cada historia de usuario y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses.

En esta fase los artefactos que se generan son el Plan de iteraciones donde los elementos a tener en cuenta durante su elaboración son: historias de usuario no abordadas, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior (al final de la última iteración el sistema estará listo para entrar en producción) y el Plan de entrega, compuesto por iteraciones de no más de tres semanas.

- **Iteraciones**

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores. Se diseñan las tarjetas CRC (Clase, Responsabilidad, Colaboración). Una tarjeta CRC representa un objeto. El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente.

Para cada una de estas iteraciones se confeccionarán las Unidades de prueba y se aprobarán las mismas antes de empezar a codificar la solución, estas ayudarán a los programadores a tener una mejor visión del comportamiento del programa. Luego se aplicarán las Pruebas de aceptación diseñadas por el cliente, destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida. El objetivo de estas pruebas es verificar el cumplimiento de los requisitos.

Producción

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, ya que realizar cambios durante esta fase implica costos, entonces se tendría que analizar la factibilidad de estos cambios.

1.6.2 Lenguaje de modelado UML como lenguaje de modelación visual

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. No es un estándar oficial, pero recibe el apoyo en gran manera por el OMG (Object Management Group). UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software actualmente UML es el estándar para el diseño orientado a objetos, debido a que es el resultado de la unión de las mejores cualidades de los otros lenguajes.[37]

1.6.3 Tecnologías relacionadas con el desarrollo de `applets`

1.6.3.1 JavaCard Runtime Environment (JCRE)

El `JCRE`[13] está comprendido por la máquina virtual de `JavaCard` (`JCVM`) además de las clases y servicios definidos en el Application Programming Interface (`API`), ambiente sobre el cual se desarrollan los `applets`. El tiempo de vida de la `JCVM`[14] es igual al tiempo de vida de la tarjeta y sus objetos mantienen sus estados entre dos sesiones con una terminal, dicho comportamiento es responsabilidad de la `JCRE`; esta es la principal diferencia entre la `JVM`[15] y la `JCVM`.

[13] Acrónimo de JavaCard Runtime Environment.

[14] Acrónimo de JavaCard Virtual Machine.

[15] Acrónimo de Java Virtual Machine.

Otras diferencias entre ellas son las limitaciones en los tipos de datos manejados y los requerimientos de hardware para la ejecución. Para investigar y conocer el funcionamiento de una `JavaCard`, hay que tener en cuenta que al realizar la especificación de la plataforma, Sun se apegó al estándar ISO 7816, el cual establece, entre otras cosas, la forma de comunicación entre una Tarjeta Inteligente y una terminal. De acuerdo al ISO 7816, el intercambio de información y comandos entre la tarjeta y el terminal se realiza a través de APDUs (Application Protocol Data Units), estos son paquetes de información con un formato específico. De acuerdo al estándar, las Tarjetas Inteligentes nunca inician la comunicación con el terminal, sino que sólo responden a los comandos que éste les envía. Se puede decir que un `applet` comienza su ciclo de vida al ser correctamente cargado en la memoria de la tarjeta, link – editada y preparada para su correcta ejecución. Al registrarse en el JCRE unen condiciones de ejecutar. El `Applet` posee el mismo tiempo de vida que la tarjeta. La clase javacard.framework.Applet, es una clase abstracta provista en el desarrollo utilizado (Developer Suite), donde se definen cuatro métodos públicos que son utilizados por el JCRE para hacer funcionar las aplicaciones.[38]

1.6.3.2 Lenguaje JavaCard

Lenguaje de programación JavaCard

Es una combinación del lenguaje `Java` con un entorno de ejecución para tarjetas inteligentes, permite ejecutar applets en el `chip` embebido en la tarjeta, los cuales contienen funcionalidades que son reutilizables para otros componentes, se ejecutan e interactúan en todo momento con el entorno de ejecución que contiene la máquina virtual de `JavaCard`, junto a las clases y servicios definidos en las API de estas.

Ambiente de ejecución de aplicaciones en las tarjetas inteligentes: JCRE (JavaCardRuntime Environment) es el ambiente sobre el cual se ejecutan los applets y comprende el `JavaCard` API y la JCVM (JavaCard Virtual Machine). La JCVM se diferencia principalmente de una JVM (Java Virtual Machine) normal en que el tiempo de vida de la misma es igual al tiempo de vida de la tarjeta.[39]

1.6.4 Tecnologías propuestas para el desarrollo del `middleware`.

1.6.4.1 Microsoft .Net Framework

Capítulo 1: Fundamentación teórica.

Es el conjunto de nuevas tecnologías que se crearon con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente, comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados. Sobre la infraestructura del `Framework` de `.Net` se reúne una serie de lenguajes y servicios que posibilitan simplificar el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables. Organiza toda la funcionalidad del sistema operativo en un espacio de nombres jerárquicos de forma que a la hora de programar resulta bastante sencillo encontrar lo que se necesita. Entre todas sus ventajas se destacan:[40]

- **Código administrado:** el Tiempo de ejecución del Lenguaje Común (CLR, por sus siglas en inglés Common Language Runtime) realiza un control automático del código para que este sea seguro, posibilitando que la aplicación se ejecute correctamente.
- **Interoperabilidad multilenguaje:** se puede escribir el código en cualquier lenguaje compatible con .Net porque siempre se compila en código intermedio o Microsoft Intermediate Lenguaje (MSIL).
- **Compilación just-in-time:** el compilador JIT (Just In Time, nombre que recibe ese tipo de compilación porque se realiza en tiempo de ejecución) incluido en el `Framework` compila el código intermedio (MSIL) de esta manera genera el código máquina propio de la plataforma. Se aumenta así el rendimiento de la aplicación al ser específico para cada plataforma.
- **Despliegue**: el desarrollo de aplicaciones distribuidas y el mantenimiento de las mismas resulta mucho más fácil mediante los. El `Framework` realiza esta tarea de forma automática mejorando el rendimiento y asegurando el funcionamiento correcto de todas las aplicaciones.
- **Documentación**: esta plataforma ofrece mucha documentación de ayuda (herramientas, `debuggers`, editores) incluida en la IDE y de soporte. Esto simplifica el desarrollo y la implementación.
- **Rendimiento**: todos los códigos que se ejecutan en el ambiente .NET son compilados, lo cual proporciona un gran rendimiento a diferencia de versiones interpretadas.
- **Rápido aprendizaje por parte de los desarrolladores:** es sencillo de aprender por la documentación y el soporte de ayuda.

- **Movilidad**: las aplicaciones pueden ser desplegadas en una amplia variedad de dispositivos.
- **Escalabilidad y flexibilidad:** escalabilidad es la capacidad de un sistema para soportar más carga de trabajo, usualmente debida al aumento de usuarios que lo utilizan. .NET ofrece métodos de escalabilidad como la carga balanceada que permite a un `cluster` de servidores (varios servidores) colaborar y dar un servicio de forma simultánea. En cuanto a la flexibilidad, el modo de programación que se emplea permite agregar nuevos módulos sin modificar la aplicación en su totalidad.

1.6.4.3 Lenguaje de programación Csharp

Aunque para la plataforma .NET es prácticamente posible programar en cualquier lenguaje, El lenguaje C# es diseñado por Microsoft para programar en la plataforma .NET, esto permite que sea mucho más sencillo que con cualquier otro lenguaje. Entre sus principales características se destacan:[41]

- **Sencillez:** C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. El código escrito en C# es auto contenido, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera. Su código es fácilmente portable debido a que el tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile.
- **Orientación a componentes:** la propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente propiedades (similares a campos de acceso controlado), eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros).
- **Eficiente:** a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros. Para ello basta marcar regiones de código como inseguras (modificador `unsafe`) y podrán usarse en ellas punteros de forma similar a cómo se hace en C++, siendo esto excelente para situaciones donde se necesite una eficiencia y velocidad de procesamiento muy grandes. Aunque en un

principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros.

1.6.5 Herramientas propuestas para el desarrollo

1.6.5.1 Altova Umodel

UModel es usado para la creación e interpretación de diseños software con el estándar UML 2.1. Es útil para el dibujo del diseño de la aplicación y puede generar código para Java o C# a partir de planos, permite realizar ingeniería inversa de programas existentes a diagramas UML claros y precisos para abarcar rápidamente su arquitectura software. Además se puede corregir el código generado o los modelos y completar la ronda produciendo automáticamente nuevos diagramas o regenerando el código.[42]

1.6.5.2 Developer Suite

Herramienta que brinda un ambiente favorable para el diseño y la implementación de `applets`, posibilita simular las funcionalidades de estos antes de ser instalados en las tarjetas inteligentes.[43]

1.6.5.3 Tecnología JavaCard

Es una tecnología que permite ejecutar de forma segura pequeñas aplicaciones Java (`applets`) en tarjetas inteligentes y similares dispositivos empotrados. `JavaCard` brinda al usuario la capacidad de programar aplicaciones que se ejecutan en la tarjeta, de modo que ésta tenga funcionalidades prácticas, en un dominio de aplicación específico. Se usa ampliamente en las tarjetas SIM (utilizadas en teléfonos móviles GSM) y en tarjetas monedero electrónico. A nivel de lenguaje, `JavaCard` es un subconjunto de `Java`: todas las construcciones del lenguaje `JavaCard` existen en Java y se comportan de la misma manera. Esto va hasta el punto de que, como parte de un ciclo estándar de desarrollo, un `Applet JavaCard` se compila en un archivo de clase `Java`(`.class`) por un compilador Java normal, sin ningún tipo de opción especial (aunque el fichero compilado será procesado posteriormente por herramientas específicas para la plataforma `JavaCard`).[39]

1.6.5.4 Visual Studio Team System

Visual Studio Team System es una plataforma de trabajo diseñada por Microsoft para que los desarrolladores de software puedan contar con un ambiente de trabajo creado a la medida de sus necesidades. Este programa, por un lado, integra una completa plataforma de herramientas y por otro, brinda un conjunto de utilidades, mediante el cual los desarrolladores de software, que se encuentran dispersos por el mundo cuentan con un único lugar de trabajo. El mismo nos brinda utilidades capaces de desarrollar componentes de software, testearlos, hacer correcciones, colocarlos en un paquete, ponerlos a la venta, etc., todo ello mediante una visión horizontal de negocio y crecimiento, que necesita de la colaboración de todos los miembros del equipo. El cual cuenta con vínculos para acoplar otros componentes de software Microsoft, como por ejemplo Office, Visual Basic, o .NET Framework, por sólo citar algunos.[44]

Conclusiones

Este capítulo centra su función tanto en un análisis acerca de las tarjetas inteligentes usadas para los sistemas de control de acceso, como las tecnologías y herramientas más óptimas para uso e implementación de las mismas. Se abordan especificaciones de las diferentes tecnologías que se utilizan con los sistemas de control de acceso basados en tarjetas inteligentes. De esta manera se puede analizar cuáles tecnologías deben usarse y qué estándares permiten realizar el modelado y la implementación de la solución propuesta, profundizando en las potencialidades que brindan. Para la elaboración del sistema se plantea el uso de algunas herramientas que proporcionan seguridad y sencillez a la aplicación.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN.

2.1 Introducción

Este capítulo propone una solución al problema científico utilizando la metodología de desarrollo de software XP[16], la misma pertenece al grupo de metodologías ágiles, es apropiada para entornos volátiles y adaptables a los cambios por lo que esto produce una reducción de costos.

De las cuatro fases de XP este capítulo elabora la solución, mediante el uso de las dos iniciales: Planificación y Exploración, en las cuales se generan diferentes artefactos como las Historias de Usuarios.

El objetivo que se persigue con la elaboración de este capítulo es mostrar la evolución de la solución durante las fases iniciales de Planificación y Exploración, además de presentar los diferentes artefactos generados en las mismas, los cuales serán premisas cruciales para la entrega final del software.

2.2 Propuesta de solución

2.2.1 Modelo de dominio

La creación de un modelo del dominio posibilita identificar conceptos asociados con el entorno en que se desarrolla la solución y mostrar las relaciones entre ellos. En el siguiente modelo se muestran las relaciones que existen entre los principales conceptos que componen el Sistema de Control de Acceso, destacándose los distintos subsistemas y los aspectos organizativos que deben ser tomados en cuenta para su desarrollo.

[16] Extreme Programming por sus siglas en inglés.

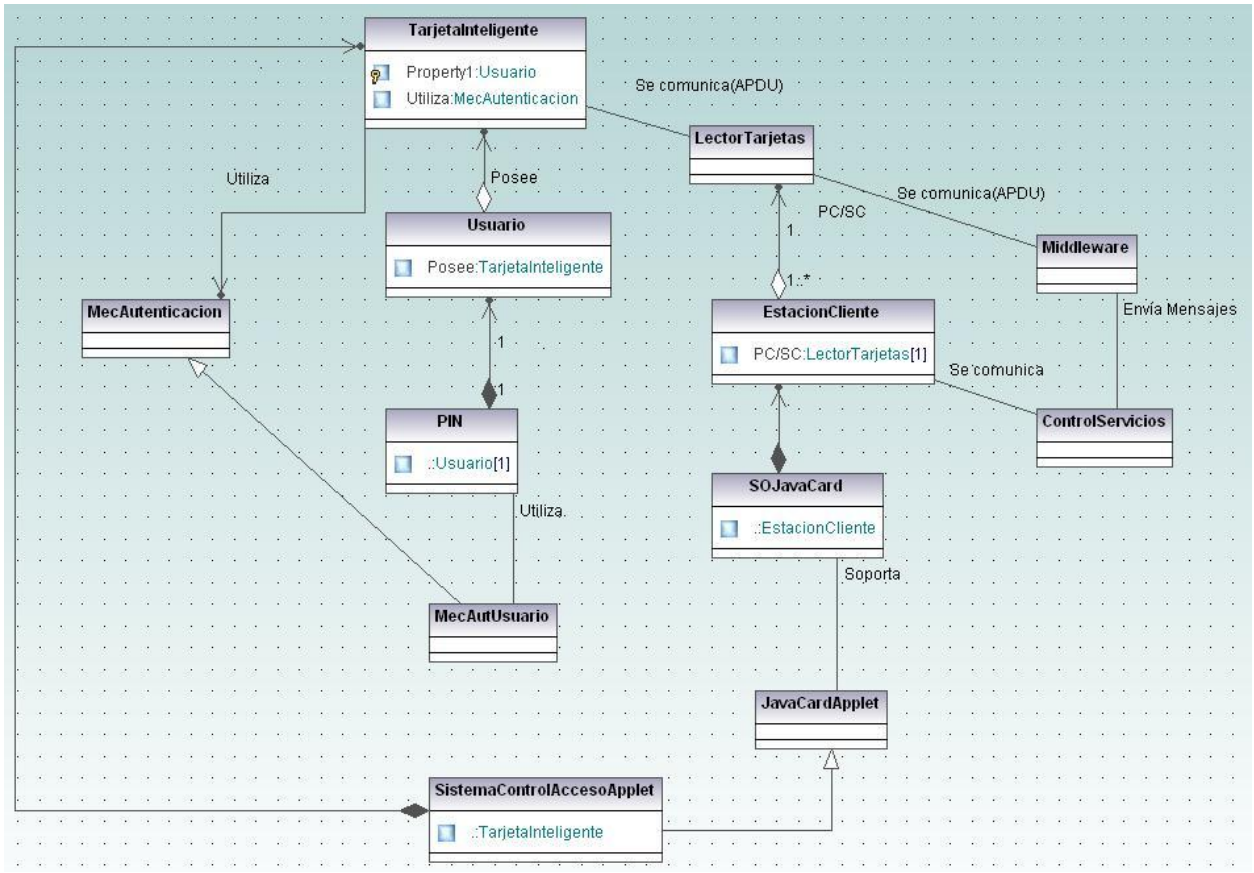

Figura 4: Diagrama de clases del modelo de dominio.

2.2.2 Conceptos asociados al modelo del dominio

Usuario: persona portadora de la tarjeta inteligente.

LectorTarjetas: es un lector compatible con el estándar PC/SC, el cual sirve de mediador para la comunicación entre la estación cliente y la tarjeta inteligente.

TarjetaInteligente: dispositivo similar en tamaño y forma a las tarjetas de crédito, tiene un circuito integrado, el mismo puede ser de sólo memoria o contener un microprocesador (CPU) con un sistema operativo que le permita una serie de funcionalidades como almacenar información, encriptar información, leer y escribir datos; similar a un ordenador.

SO `JavaCard`: la tecnología `JavaCard` combina parte del lenguaje de programación Java con un entorno de ejecución optimizado para tarjetas inteligentes y similares.

`Applet`: es una aplicación instalada dentro de la tarjeta inteligente que permite gestionar la información que se almacena en ella.

`Middleware`: software de conectividad que funcionan como una capa de abstracción, usado para interactuar con la tarjeta inteligente.

Mecanismo Autenticación: mecanismo de autenticación común.

Mec.Aut.Tarjeta: mecanismo que utiliza la tarjeta para establecer un canal seguro con el lector.

Mec.Aut.Usuario: mecanismo de autenticación que sirve al sistema para comprobar que existe un lazo seguro entre el usuario y la tarjeta inteligente, puede ser a través de la introducción del PIN personal, huella digital, lectura de iris, entre otros. Permite comprobar que la tarjeta pertenece a la persona que intenta utilizarla.

PIN: (Personal Identification Number o Número de Identificación Personal en español) es un valor numérico usado para identificarse y poder acceder a la información de la tarjeta.

Estación Cliente: computadora que utiliza el usuario.

Control Servicio: es un punto en una computadora donde se instalan todas las condiciones para poder interactuar con la solución, para el control de acceso a la información de las tarjetas. Entre los elementos indispensables también podemos citar, los lectores PC/SC, entre otros.

2.3 Historias de Usuario

En el contexto de XP las historias de usuario son utilizadas para especificar los requisitos funcionales del software desde el punto de vista del cliente. Se asignan al desarrollador encargado de la programación con un número de horas de desarrollo estimado, el cliente y el analista en conjunto, las describen; las historias de usuario guían la construcción de los test de aceptación (casos de prueba).

Historia de usuario	
Número: HU_1	**Nombre de Historia de Usuario:** Seleccionar lector con el cual se va a establecer la comunicación con la tarjeta inteligente.
Modificación de Historia de Usuario Número: Ninguna	
Usuario:	**Iteración Asignada:** 1
Prioridad en negocio: Alta	**Puntos Estimados:** 1
Riesgo en desarrollo: Medio	**Puntos Reales:** 1
Descripción: El sistema le brinda una relación de los lectores conectados a la computadora, dándole la posibilidad de escoger uno para establecer la conexión con la tarjeta, sino el sistema escoge el primero de la lista por defecto.	
Observaciones: En caso de que no exista ningún lector conectado al ordenador, el sistema muestra un aviso.	

Tabla 3. HU_1 Seleccionar lector con el cual se va a establecer la comunicación con la Tarjeta Inteligente.

Historia de usuario	
Número: HU_2	**Nombre de Historia de Usuario:** Gestionar comunicación con la tarjeta inteligente.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario:	**Iteración Asignada:** 1
Prioridad en negocio: Alta	**Puntos Estimados:** 3
Riesgo en desarrollo: Medio	**Puntos Reales:** 3
Descripción: El sistema intentará acceder a la aplicación, para ello se debe establecer una conexión con la tarjeta que permitirá iniciar la comunicación de la aplicación. Una vez establecida la conexión con el lector, el sistema le notificará al usuario a través de un mensaje si la conexión con la tarjeta se ha establecido o no. **Esta tiene dos estados posibles:** Conectada en el lector. Desconectada del lector. **Existen varios casos en que la conexión con la tarjeta puede ser interrumpida:** Si el usuario extrae la tarjeta del lector. Si el usuario desconecta el lector de la computadora. Si el usuario cierra la conexión a través de la aplicación.	

Observaciones: Si no existiera ningún lector conectado a la computadora el sistema notificaría un mensaje de error.

Tabla 4. HU_1 HU_2 Gestionar comunicación con la tarjeta inteligente.

Historia de usuario	
Número: HU_3	**Nombre de Historia de Usuario:** Establecer canal seguro de comunicación con la tarjeta, siguiendo `Global Platform`.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario:	**Iteración Asignada:** 1
Prioridad en negocio: Alta	**Puntos Estimados:** 3
Riesgo en desarrollo: Medio	**Puntos Reales:** 3
Descripción: Se establece un canal de intercambio de información de forma segura entre el `Middleware` y el `Applet` de Control de Acceso, utilizando Protocolo de Canal Seguro "01" según especificaciones de `Global Platform`.	
Observaciones:	

Tabla 5. HU_3Establecer canal seguro de comunicación con la tarjeta, siguiendo `Global Platform`.

Historia de usuario	
Número: HU_4	**Nombre de Historia de Usuario:** Permitir autenticación del usuario por Número de Identificación Personal (PIN).
Modificación de Historia de Usuario Número: Ninguna	
Usuario:	**Iteración Asignada:** 1
Prioridad en negocio: Alta	**Puntos Estimados:** 1
Riesgo en desarrollo: Alto	**Puntos Reales:** 1
Descripción: Consiste en realizar la autenticación del Usuario portador de la tarjeta, introduciendo este su número de PIN para validar la autenticidad de posesión de la tarjeta.	
Observaciones: En caso de que la autenticación haya sido incorrecta se notifica dicha información enviando un mensaje	

de error, si es correcta se notifica que ha sido satisfactoria.

Tabla 6. HU_4 Permitir autenticación del usuario por Número de Identificación Personal (PIN).

Historia de usuario	
Número: HU_5	**Nombre de Historia de Usuario:** Cambiar número de identificación personal (PIN) **.**
Modificación de Historia de Usuario Número: Ninguna	
Usuario:	**Iteración Asignada:** 1
Prioridad en negocio: Alta	**Puntos Estimados:** 1
Riesgo en desarrollo: Alto	**Puntos Reales:** 1
Descripción: Una vez establecida la comunicación con la tarjeta, el usuario escoge la opción de cambiar PIN. Introduce el actual a modo de verificación y luego introduce el nuevo PIN, el sistema verifica el PIN que aún está en uso y de ser válido procede al cambio.	
Observaciones: En caso que el PIN actual no sea el correcto, se cancela la operación y se le informa al usuario.	

Tabla 7. HU_5 Cambiar Número de Identificación Personal (PIN).

Historia de usuario	
Número: HU_6	**Nombre de Historia de Usuario:** Verificar condiciones de acceso a la información dentro del `Applet` de Control de Acceso.
Modificación de Historia de Usuario Número: Ninguna	
Usuario:	**Iteración Asignada:** 1
Prioridad en negocio: Alta	**Puntos Estimados:** 2
Riesgo en desarrollo: Medio	**Puntos Reales:** 2
Descripción: Se verifican las condiciones de acceso de un certificado emitido por una autoridad certificadora referente a una terminal de trabajo del `Applet` de Control de Acceso. El `Applet` comprueba que las condiciones de acceso son las correctas para realizar la solicitud de gestión de la información. **Información:** En los atributos extendidos del certificado se encuentran los tipos de Modos de Acceso que se pueden tener:	

- Solo lectura de la información. - Solo escritura de la información. - Lectura / Escritura de la información.
Observaciones: Si las condiciones de acceso no son cumplidas el `Applet` retorna un mensaje `APDU` notificando el error el cual está definido dentro del estándar `Global Platform`.

Tabla 9. HU_6 Verificar condiciones de acceso a la información dentro del `Applet` de Control de Acceso.

Historia de usuario	
Número: HU_7	**Nombre de Historia de Usuario:** Gestionar datos en la tarjeta.
Modificación de Historia de Usuario Número: Ninguna	
Usuario:	**Iteración Asignada:** 2
Prioridad en negocio: Alta	**Puntos Estimados:** 3
Riesgo en desarrollo: Medio	**Puntos Reales:** 3
Descripción: La gestión de datos en la tarjeta está conformada por un primer proceso el cual se encarga de personalizar y leer los datos de la tarjeta, donde obtiene los datos de la persona y los almacena; el otro proceso es el que lee dichos datos y se encarga de visualizarlos en una interfaz.	
Observaciones:	

Tabla 10. HU_7 Gestionar datos en la tarjeta.

2.4 Requerimientos no funcionales

Requerimientos mínimos de hardware:

- Lector de tarjetas incorporado a la PC que cumpla con el estándar PC/SC versión 1.0 o superior.

Requerimientos de software:

- Se requiere que estén instalados en la PC cliente los drivers del lector de tarjetas.

Usabilidad:

- La aplicación deberá permitir la incorporación de nuevas funcionalidades.

Portabilidad:

- La aplicación debe ser compatible con cualquier lector de tarjetas que cumpla con el estándar PC/SC.

Seguridad:

- **Confiabilidad:**

✓ Debe recuperarse en el menor tiempo posible en caso de producirse una falla.

La información almacenada en el `applet` estará protegida de ataques externos a través de la seguridad que define el proveedor de tarjetas, su sistema operativo y la tecnología `JavaCard`. La información contenida en las tarjetas estará protegida por la seguridad que defina el proveedor de las mismas y por el canal seguro establecido previo a su comunicación.

- **Confidencialidad:**

✓ La aplicación soportará mecanismos de seguridad comunes entre la tarjeta y el `middleware`.

- **Integridad:**

✓ La información contenida en la tarjeta, será objeto de cuidadosa protección contra la corrupción y estados inconsistentes.

2.5 Metáfora

Una metáfora para el sistema es una vía fácil y concreta de explicar cómo funciona el mismo, debe tener un contenido que sea suficiente para guiar la arquitectura del proyecto. Es necesario que los nombres de los objetos con los que se trabaja en el sistema tengan relación con los mismos.

La solución para gestionar el control de acceso a instalaciones utilizando tarjetas inteligentes puede utilizarse en cualquier instalación que lo requiera. Está conformada por un `applet` que está contenido dentro de la tarjeta y un `middleware` encargado de la interacción con el mismo. El sistema confirmará que el usuario es el real portador de la tarjeta a través de la comparación del PIN introducido por este y el que se encuentra almacenado en la tarjeta, a partir de entonces se procede a la obtención de los datos almacenados en la misma. La instalación podrá hacer uso del sistema para varias aplicaciones dentro de una misma instalación.

2.6 Arquitectura

La siguiente propuesta del sistema de control de acceso basa su implementación en una arquitectura simple cliente/servidor conocida como arquitectura de 2 capas. Teniendo en cuenta las características del sistema, cuyo principal papel es mediar entre la aplicación a desarrollar y el lector de la tarjeta inteligente, se identifica una capa estación (cliente) que contiene la aplicación la cual implica al `middleware`. En segundo lugar se tiene la capa Lector (servidor), donde se encuentra el applet el cual contiene toda la información personal del portador de la tarjeta inteligente.[45]

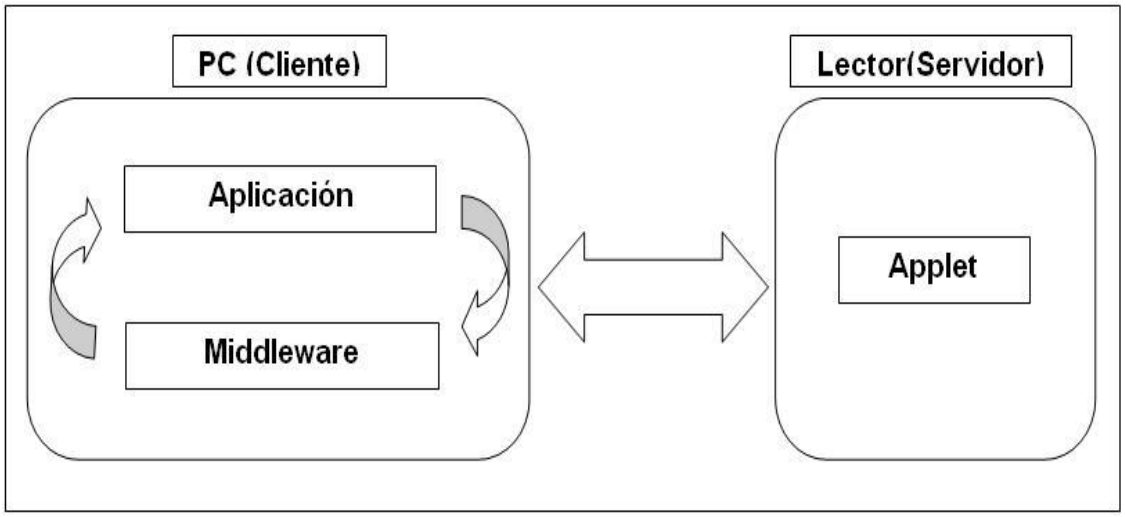

Figura 5: Componentes de la arquitectura.

Las arquitecturas en capas son muy utilizadas para el desarrollo de aplicaciones en la actualidad por las grandes ventajas que proporcionan. El principal objetivo que persigue es reducir dependencias entre artefactos, situándolos en capas lógicas, donde cada capa depende del servicio prestado por la inferior y presta un servicio a la superior, proporcionando a los desarrolladores ventajas en cuanto al mantenimiento y reutilización de estos componentes.

A continuación se presenta la arquitectura definida en tres capas del sistema:

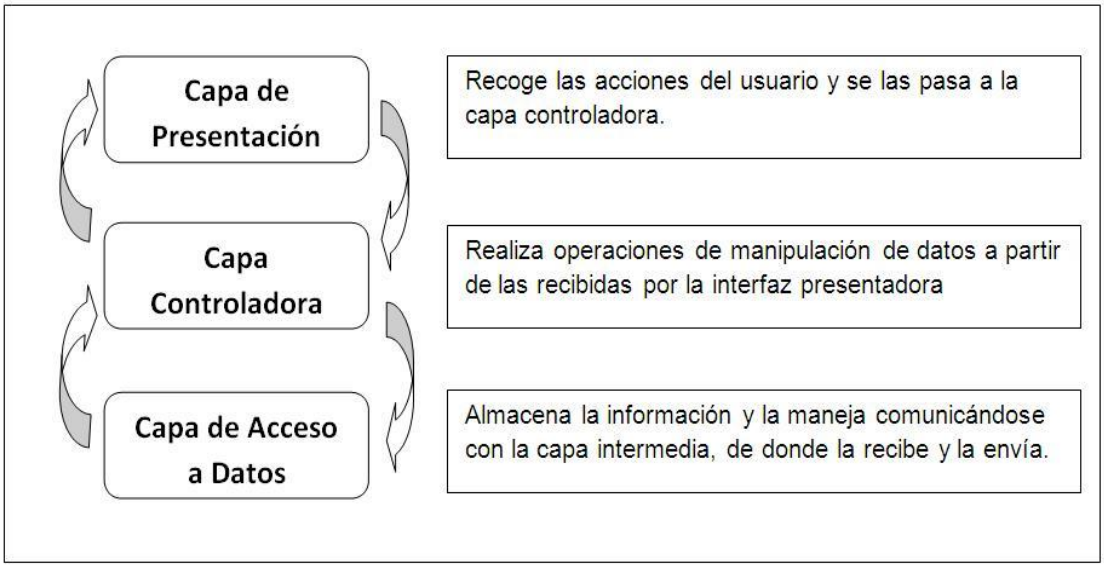

Figura 6: Diagrama de Arquitectura.

Capa de presentación: esta es la capa que interactúa con el usuario, es la encargada de modelar como se recogerán y mostrarán los datos del proceso asistencial, así como de la apariencia que tendrá la interfaz visual. Esta se comunica con la capa controladora o `middleware` a la cual envía todas las solicitudes y las muestra cuando estas hayan sido procesadas. En la solución desarrollada, esta capa se llamará: `test Middleware`.

Capa controladora: la capa controladora representa las clases del sistema, es la encargada de darle solución a las historias de usuario, en esta es donde se implementan las restricciones que deberá cumplir la solución. Se comunica con la capa de presentación para recibir las solicitudes del usuario, las envía a la capa de acceso a datos y envía las respuestas a la capa de presentación después del procesamiento de la solicitud. En el sistema que será desarrollado posteriormente esta capa se llamará: `Middleware`.

Capa de acceso a datos: esta capa contendrá una aplicación donde estará almacenada la información referente al portador de la tarjeta inteligente y esta será la encargada del manejo de dicha información. La misma se comunicará con la capa intermediaria, de donde recibe la solicitud de guardar información o demostrarla. En la solución desarrollada, esta capa se llamará: `SistemaControlAccesoApplet`.

2.6.1 Patrones del diseño.

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software, que están sujetos a contextos similares. Se debe tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios). Los mismos están divididos en 2 grupos los del grupo **GoF**[17] y los del grupo **GRASP**[18]**. [46]**

Los patrones de diseño el grupo de **GoF** clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento.[47]

Creacionales: tratan las formas de crear instancias de objetos. El objetivo de estos patrones es abstraer el proceso de instanciación, ocultar los detalles de cómo los objetos son creados o inicializados.

Estructurales: los patrones estructurales describen las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionados, pueden ser incluso, objetos simples u objetos compuestos.

Comportamiento: los patrones de comportamiento ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos.

En la implementación de la solución desarrollada, se utilizó el patrón que se destacan a continuación:

Método de fabricación: el patrón se aplica centralizando la solución en una clase constructora (`ClienteMiddlewareAppletCA`), la cual utilizará la creación de objetos de un subtipo, de un tipo determinado, ocultando al usuario la casuística para elegir el subtipo que crear.

Los patrones del grupo **GRASP** en diseño orientado a objetos, son patrones generales de software para asignación de responsabilidades, aunque se considera que más que patrones

[17] Gang of Four por sus siglas en ingles.
[18] General Responsibility Assignment Software Patterns por sus siglas en ingles.

propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software.

En la solución desarrollada, se referencian varios patrones de este grupo, entre los que destacan:

Experto: experto en información es el principio básico de asignación de responsabilidades. A través del mismo se indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo, este caso la clase controladora `ClienteMiddlewareAppletCA`. De este modo se obtendrá un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento).

Creador: el patrón creador ayuda a identificar quién debe ser el responsable de la creación (en este caso, la clase `Util`[19]) de nuevos objetos o clases, la cual contiene la información necesaria para realizar la creación del objeto, además usar directamente las instancias creadas del objeto.

Controlador: el mismo se utiliza como intermediario, entre una determinada interfaz (`testMiddleware`) y el algoritmo que la implementa, de tal forma, que es la que recibe los datos del usuario y la que los envía a las distintas clases (`Util`, `Usuario`, `ClienteMiddlewareAppletCA`), según el método llamado. El cual se puede aplicar perfectamente debido a la condición, de que lógica de negocios, debe estar separada de la capa de presentación, para aumentar la reutilización de código y a la vez tener un mayor control.

Bajo acoplamiento: de esta forma se cuenta con clases, ligadas entre sí, con la menor dependencia posible. De tal forma, en caso de producirse una modificación en alguna de ellas, se tiene la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

[19]Util es una clase, de la solución desarrollada en Visual Studio 2005, por lo que al referirse a ella, no se le pone tilde y se le hace un cambio de tipo de letra, debido a que se toma como palabra técnica.

2.7 Plan de Entrega

Luego de elaborar las Historias de Usuario, se realiza el Plan de entrega donde se estima el tiempo de desarrollo de las HU, para marcar cuanto tiempo se demora la implementación de cada una y por supuesto la fecha de "salida" de las versiones funcionales del producto. (Ver Anexo 3)

2.8 Estimación de Tiempo

El tiempo necesario para desarrollar cada historia de usuario, es estimado por los programadores. El valor del mismo se expresa en semanas y a medida que transcurren las iteraciones este se va tornando real. (Ver Anexo 4).

2.9 Plan de Iteraciones

Como parte del ciclo de vida de un proyecto usando la metodología XP se crea el plan de duración de cada una de las iteraciones que se han definido, que tiene como objetivo mostrar la duración y el orden en que serán implementadas las historias de usuario dentro de cada iteración. Para la solución se han definido 8 historias de usuario divididas en 2 iteraciones, de acuerdo a los intereses del cliente, para una duración total del proyecto de 17 semanas. (Ver Anexo 5)

2.10 Estudio de Factibilidad

Dentro de la planificación de proyectos de software, la estimación es la tarea de mayor importancia, en la misma se determinan, los recursos de hardware y software, costo, tiempo y esfuerzo. Para lo cual se utilizó COCOMO II como modelo de estimación de costos, el cual posibilita realizar estimaciones en dependencia del tamaño del software y de otros factores como la escala, el costo, este último describe detalles dela naturaleza del producto, hardware utilizado, personal involucrado y características propias del proyecto. (Ver Anexo 6)

2.10.1 Estimación de esfuerzo

COCOMO II está compuesto por tres modelos denominados: Composición de aplicación, Diseño temprano y Post-arquitectura. Surgen en respuesta a la diversidad del mercado actual y futuro de desarrollo de software. Debido a las necesidades y características de la solución, se ha utilizado una combinación de los dos últimos.

Capítulo 2: Propuesta de Solución.

El modelo de Diseño temprano ajusta el esfuerzo nominal usando siete factores de costo.[48](Ver Anexo 7)

La fórmula para el cálculo de la estimación del esfuerzo es la siguiente:

$$\text{PM}_{\text{estimado}} = \text{PM}_{\text{nominal}} \text{ X } \prod_{i=1}^{7} EMi$$

Dónde:

$$\text{PM}_{\text{nominal}} = \text{A X } (KSLOC)^{B}$$

Dónde:

PM estimado: es el esfuerzo nominal ajustado por 7 factores, que reflejan otros aspectos propios del proyecto que afectan al esfuerzo necesario para la ejecución del mismo, expresado en meses/personas.

KSLOC: es el tamaño del software a desarrollar expresado en miles de líneas de código fuente.

A: es una constante que captura los efectos lineales sobre el esfuerzo de acuerdo a la variación del tamaño, (A= 2.94).

B: es el factor exponencial de escala, toma en cuenta las características relacionadas con las economías de escala producidas cuando un proyecto de software incrementa su tamaño.

EMi: corresponde a los factores de costo que tienen un efecto multiplicativo sobre el esfuerzo, llamados multiplicadores de esfuerzo.

Los modelos de estimación de costos del software a menudo tienen un factor exponencial para considerar los gastos y ahorros relativos de escala encontrados en proyectos de software de distinto tamaño. B se usa para capturar estos efectos y es calculado con la ecuación:

$$\text{B} = 0.91 + 0.01 \text{ x } \sum_{j=1}^{5} \text{SF}_j$$

Si B < 1.0 El proyecto presenta ahorros de escala.

Si B = 1.0 Los ahorros y gastos de escala están equilibrados.

Si B > 1.0 El proyecto presenta gastos de escala.

Un factor de escala de un proyecto, **SFj**, se calcula sumando todos los factores y se usa para determinar el exponente de escala, B.

Sustituyendo valores en la fórmula anterior queda que B = 1.0778.

2.10.2 Puntos función

La fórmula de Albretch (Albretch, 1979) para calcular los puntos función, es la siguiente:

FP = UFP X TCF

Dónde:

FP: puntos función

UFP: puntos función no ajustados

TCF: factor de complejidad técnica

Para calcular los UFP, se deben identificar los siguientes ítems: entradas externas, salidas externas, archivo lógicos internos, archivos externos de interfaces, solicitudes externas.

Luego se clasifican de acuerdo al grado de complejidad en: simple, medio o complejo. Se asigna un peso a cada uno según el tipo y el grado de complejidad correspondiente. (Ver Anexo 8)

Finalmente los UFP son calculados mediante la sumatoria de los pesos de todos los ítems identificados.

$$\mathrm{UFP} = \sum_{i=1}^{15} (\mathrm{Cantidad\_Items\_Tipo}_{\,i})\ \mathrm{X}\ (\mathrm{Peso})$$

Para el cálculo del Factor de complejidad técnica, TCF, se considera la siguiente fórmula:

$$TCF = 065 + 0.01 \text{ X } \sum_{i=1}^{14} Fi$$

Donde los Fi corresponden a los pesos asignados a factores, como reusabilidad, comportamiento, complejidad, confiabilidad, etc., que ya están contemplados por COCOMO II a través de los factores de costo. Por lo que este modelo utiliza los UFP como métrica de determinación de tamaño.

Despejando de la fórmula original se tiene un total de 50 de Puntos función sin ajustar (UFP) equivalente a los Puntos función (FP).

2.10.3 Líneas de código

El tamaño de una aplicación se mide en unidades de líneas de código fuente (KSLOC), este valor en COCOMO II puede estimarse a partir de Puntos función sin ajustar convirtiendo a SLOC y luego dividiendo por 1000.

Si se opta por utilizar los Puntos de función sin ajustar para determinar el tamaño del proyecto, deben convertirse a líneas de código fuente en el lenguaje de implementación (ensamblador, lenguajes de alto nivel, lenguajes de cuarta generación, etc.) para evaluar la relativamente concisa implementación por Puntos función.(Ver Anexo 9)

Una vez calculados los elementos necesarios, se procede al cálculo del Esfuerzo nominal del proyecto con la fórmula anteriormente planteada, quedando:

$$PM_{\text{nominal}} = 2.94 \text{ x } 1.450^{1.0778}$$

Derivándose del resultado anterior se encuentra el esfuerzo estimado del proyecto quedando:

PM estimado = 4.38 x 0.80

PM estimado = 3.50 meses/persona

2.10.4 Estimación de tiempo de desarrollo

La ecuación inicial para los tres modelos de COCOMO II es:

$$TDEV = [3.67 \text{ X } PM^{(0.28+0.2X(B-1.01))}] \text{ X } \frac{SCED\%}{100}$$

Dónde:

TDEV: es el tiempo en meses que transcurre desde la determinación de los requerimientos a la culminación de una actividad que certifique que el producto cumple con las especificaciones.

PM: es el esfuerzo expresado en meses/personas, calculado sin tener en cuenta el multiplicador de esfuerzo SCED.

B: es el Factor de escala.

SCED%: es el porcentaje de compresión/expansión del cronograma.

Aplicando la fórmula anterior queda:

TDEV = [3.67 x 3.50 (0.28+0.2*(B-1.01))]*SCED%/100

TDEV = 5.3 meses.

El tiempo y esfuerzo de desarrollo se consideran mucho más próximos a la realidad al utilizar el modelo COCOMO II, se destaca el hecho de que para proyectos pequeños resulta efectivo el lograr un prototipo rápido como técnica de cumplimiento a los requisitos, que si bien al inicio demanda de un esfuerzo y tiempo adicional, esto se ve claramente compensado al momento de desarrollar el producto final.

2.11 Costos y beneficios.

El desarrollo de un producto siempre trae aparejado un costo de producción, el cual debe ser justificado de acuerdo con los beneficios que el mismo reporta. La solución propuesta no incurre en grandes gastos, por lo que se concluye que su desarrollo es factible.
El sistema de control de acceso no se ha concebido con fines comerciales inmediatos, pero es una solución independiente, que puede comercializarse en el futuro, reportando beneficios monetarios al Centro de Identidad y Seguridad Digital y en general a la Universidad de Ciencias Informáticas y al país.

Capítulo 2: Propuesta de Solución.

Conclusiones

- En el presente capitulo se determinaron las bases del porqué la elección de un modelo de dominio, el cual agruparía todos los conceptos asociados a nuestra solución así como las relaciones entre estos conceptos, que son determinados luego del estudio de las herramientas, tecnologías y elementos tangibles asociados al dominio de la solución.
- Se definieron los requerimientos funcionales que caracterizan a la solución así como los requerimientos no funcionales que brindarán las cualidades que se deben de tener en cuenta para desarrollar una solución adecuada.
- Los comandos APDU que son enviados al `Applet` de Control de Acceso están definidos bajo la estructura del estándar ISO / IEC 7816 – 4, manipulándose también comandos definidos dentro de este estándar internacional, así como otros comandos que son específicos de la solución que en su totalidad son el medio de transporte de información comunicativa con el `Applet` de Control de Acceso.
- Se realizó un estudio de factibilidad para estimar el esfuerzo y el tiempo que tomaría la realización de la solución, así como se analizaron los gastos en que se incurrirá con la implementación de la misma haciendo una comparación con los beneficios que reportará, llegándose a la conclusión de que es factible su desarrollo.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA.

3.1 Introducción

Este capítulo tiene el objetivo de mostrar el diseño de la solución y explicarlo, además continuar con el uso de XP mediante el análisis de sus fases de Iteraciones a primera liberación y Producción, previamente en la fase de planificación se obtuvieron las historias de usuario de las cuales se hará uso en este capítulo.

Seguido de la fase de Exploración y Planificación, XP define las fases Iteraciones a primera liberación y Producción. En la planificación se definieron las iteraciones y en cada iteración se diseñan, prueban y codifican cada una de las historias de usuario.

El objetivo que se persigue con la elaboración de este capítulo es mostrar la evolución de la solución durante las fases de Iteraciones a primera liberación y Producción. Además de explicar el diseño de la solución, así como los principales componentes y las relaciones que existen entre ellos. A través de diferentes diagramas se ofrece una panorámica del funcionamiento de la solución y cómo ocurren los principales flujos de procesos.

3.2 Iteraciones

En esta fase es donde se da cumplimento al Plan de iteraciones. En cada iteración se desarrollan las sub-fases de Diseño, realización de Pruebas unitarias, Codificación de la solución y Refactorización, todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable. Al finalizar esta fase el cliente estará listo para realizar las Pruebas de aceptación.

3.2.1 Tareas de la ingeniería

Todo el trabajo de la iteración es expresado en tareas de programación, las cuales se realizan para especificar las acciones llevadas a cabo por los programadores en cada historia de usuario, ya que éstas no ofrecen el nivel de detalle requerido para saber qué implementar.

Según el Plan de iteraciones las historias de usuario se agruparon en dos iteraciones. A continuación se muestran las tareas de ingeniería derivadas de cada historia de usuario por iteración.

Responsable	Historia de Usuario	Tarea
Oren Fornaris	Seleccionar lector con el cual se va a establecer la comunicación con la Tarjeta Inteligente.	- Listar lectores conectados.
Oren Fornaris	Gestionar comunicación con la tarjeta inteligente.	- Notificar al usuario del estado de la conexión con la tarjeta. - Establecer conexión con la tarjeta. -Cerrar conexión con la tarjeta
Kirenia Garcia	Establecer canal seguro de comunicación con la tarjeta, siguiendo Global Platform	- Establecer un canal de intercambio de información de forma segura entre el `Middleware` y el `Applet de` Control de Acceso. - Utilizar Protocolo de Canal Seguro "01".
Kirenia Garcia	Permitir autenticación del usuario por Número de Identificación Personal (PIN).	- Realizar la autenticación del Usuario portador de la tarjeta. -Validar la autenticidad de posesión de la tarjeta.
Kirenia Garcia	Cambiar Número de Identificación Personal (PIN).	- Realizar la autenticación del Usuario portador de la tarjeta. -Confirmar su número de PIN. - Cambiar PIN.
Oren Fornaris	Verificar condiciones de acceso a la información dentro del `Applet` de Control de Acceso.	- Verificar condiciones de acceso. -Comprobar que las condiciones de acceso son las correctas. - Realizar la solicitud de gestión de la información.
Oren Fornaris	Gestionar datos en la tarjeta.	-Personalizar y leer los datos de la tarjeta, -Obtener y almacenar los datos de la persona. - Leer dichos datos. - Visualizarlos en una interfaz.

Tabla 19. Distribución de las tareas de ingeniería.

Capítulo 3: Implementación y prueba.

3.3 Diseño de la solución

Luego de realizar la descripción de las tareas de la ingeniería, se hace necesario comenzar el código, por lo que el equipo de trabajo presenta las tarjetas CRC en sesiones que se realizan, dichas tarjetas contienen la cantidad de clases a implementar y las responsabilidades que estas tendrán. XP propone la puesta en práctica de ciertos principios a la hora de realizar estas sesiones de trabajo, para garantizar la agilidad en el proceso de desarrollo.

Según **Wells, J. Donovan** (Wells, 1999) estos principios son:

- **Simplicidad**: Un diseño simple siempre se termina más rápido y es más fácil de entender que uno complejo.
- **Uso de tarjetas CRC**: (clase, responsabilidad, colaboración), estas tarjetas son manejadas por el equipo de desarrollo durante la codificación de la solución; generalmente cada tarjeta representa una clase diferente en la codificación y tienen como ventaja que todo el equipo contribuye a la elaboración del diseño de la solución.
- **No adicionar funcionalidades tempranamente**: Mantener el sistema lo más separado de las funcionalidades extras que no sean imprescindibles. Solo el 10% de las funcionalidades extras son utilizadas y hacen perder el 90% del tiempo.

El diseño de la solución se realizó por iteraciones donde se confeccionaron las tarjetas CRC, cumpliendo con los principios mencionados anteriormente. (Ver Anexo 10)

3.3.1 Descripción del principal flujo de procesos

Proceso Verificar PIN

Este proceso se basa en comprobar que el PIN que un usuario introduce, sea correcto y tenga así el acceso solicitado. Para que ocurra dicho proceso tiene que existir un lector conectado y a su vez una tarjeta conectada al lector, luego es seleccionado el `applet` de control de acceso, luego se abre el canal seguro y se verifica el PIN, que es introducido por el portador de la tarjeta, el `middleware` envía los datos al `applet` de la tarjeta para que sean verificados; luego el `applet` envía la respuesta obtenida al `middleware` y este a su vez se la muestra al terminal de servicio. Si es correcto el PIN se muestra un mensaje “PIN Correcto”, de lo contrario muestra un mensaje de “PIN Incorrecto”.

Capítulo 3: Implementación y prueba.

Diagrama de secuencia, Verificar PIN.

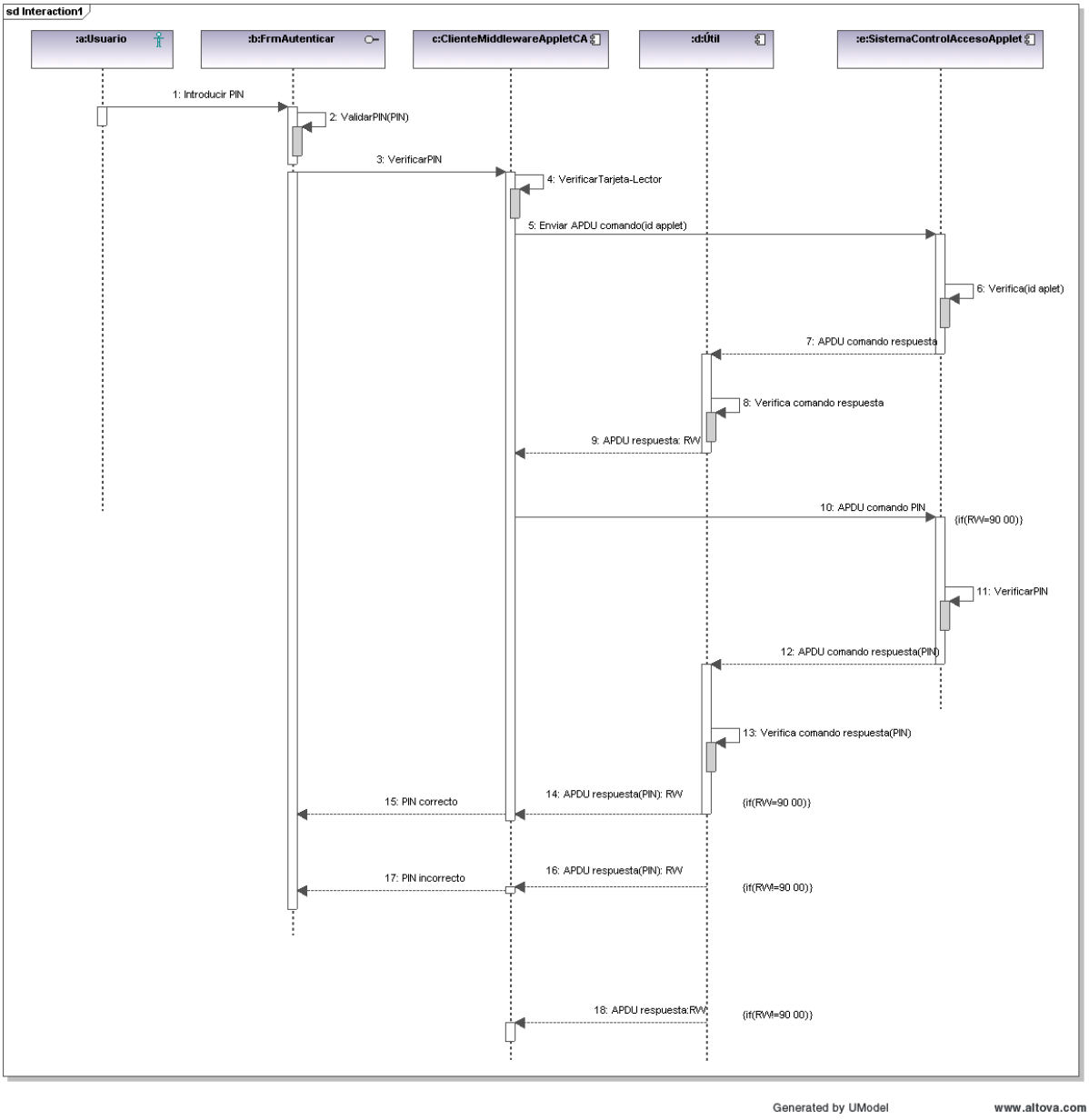

Figura 7: Diagrama de secuencia, Verificar PIN.

3.4 Fase de producción

3.4.1 Pruebas Unitarias

Las pruebas unitarias se realizaron sobre el código y el diseño con el fin de que la solución no tenga errores y el producto final tenga éxito. Dichas pruebas dada la proposición de XP deben

ser repetidas y automatizadas, asegurando con esto el funcionamiento de los componentes de manera individual antes de realizar su integración. Para realizarlas una buena práctica a tener en cuenta es la de implementar las historias de usuario luego de escribir el código. Al terminarse la codificación de las funcionalidades, se llevaron a cabo las pruebas previstas, para detectar los errores, luego de corregirlos se repitieron dichas pruebas.[49]

- En XP las pruebas del sistema se dividen en dos pequeños grupos:
- **Pruebas unitarias**: encargadas de verificar el código y diseñada por los programadores.
- **Pruebas de aceptación o pruebas funcionales**: destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.
- Las pruebas de unidad se crean por los programadores antes de empezar a codificar lo cual hará más sencillas y efectivas las pruebas finales. Se corren reiteradamente a lo largo de todo el proyecto, asegurando siempre el funcionamiento correcto de cada componente por individual antes de realizar su integración. Evitan las ambigüedades y los quedan afinados en la prueba. Una funcionalidad está terminada cuando pasa todas sus pruebas de unidad.
- Para el diseño de las pruebas existe una secuencia a seguir:
- Escribir una prueba.
- Compilar una prueba.
- Ejecutar una prueba y hacer que falle.
- Ejecutar una prueba bien.
- Refactorizar el código.

Para realizar la prueba de unidad del `middleware` se utilizó la herramienta Visual Studio Test Professional, que propone el Microsoft Visual Studio 2010. Esta herramienta proporciona una interfaz para la ejecución de pruebas de varios tipos. Con este programa se pueden crear planes de pruebas, conjuntos de pruebas y casos de pruebas con capacidades de anidamiento.

La prueba de unidad del `applet` se realizó en el entorno de desarrollo Developer Suite. El cual posee una herramienta de simulación de extremo-a-extremo, que proporciona facilidad para la descarga de los `applet` en una tarjeta. `JCard Manager` proporciona un solo clic de depuración de sus aplicaciones, para obtener como resultado un `applet` 100% funcional, cuando se descarga en una tarjeta real. Por lo tanto se arriba a las siguientes conclusiones:

- Las herramientas de pruebas están implementadas en la mayoría de los lenguajes de programación. Las pruebas son inherentes a la vida de un proyecto, debido a que posibilitan que al ser realizadas en el diseño y en el software, la solución sea exactamente como se especifican los códigos de prueba establecidos.

- Las pruebas unitarias son una herramienta muy útil en el desarrollo y diseño del software ya que ayudan a garantizar que el programa hace justo lo se especifica en los códigos de pruebas que lo definen.

Pruebas Unitarias del `applet`. (Ver Anexo 11).

Pruebas Unitarias del `middleware`. (Ver Anexo 12).

3.4.2 Pruebas de aceptación

Las pruebas de aceptación también conocidas como pruebas del cliente, son utilizadas para probar que las HU han sido implementadas de forma correcta al final de cada iteración y por lo tanto comprueban que las funcionalidades del sistema se encuentran en relación con las HU definidas. (Ver Anexo 13)

Conclusiones

Al concluir las fases de Iteraciones a primera liberación y Producción, queda demostrado:

- Realizar las tareas de la ingeniería basadas en las Historias de Usuario posibilitó que los programadores realizaran las funcionalidades específicas a implementar.

- La solución se ejecuta de manera satisfactoria durante todo el ciclo de implementación, reduciendo el tiempo de compilación-ejecución. Esto es gracias a las pruebas realizadas durante el desarrollo.

CONCLUSIONES

En el desarrollo de este trabajo, se especificó la documentación completa del sistema propuesto; abarcando desde un estudio de los sistemas que utilizan tarjetas y aplicaciones existentes, hasta el análisis y diseño del sistema de control de acceso.

- Se ha demostrado la necesidad de crear un sistema que permita restringir el acceso del personal a las instalaciones, utilizando las tarjetas inteligentes.
- El `Middleware` de Control de Acceso cuenta con un grupo de clases que le posibilitan la creación de comandos APDU para cada uno de los requerimientos de la solución. El `Middleware` utiliza el SmartCard Framework para gestionar la comunicación con las tarjetas electrónicas.
- El `Applet` de Control de Acceso cuenta con un proceso completo de verificación de las condiciones de acceso y las condiciones de seguridad que se necesitan.
- El sistema cumple con los requerimientos establecidos para el desarrollo del mismo, brindando siempre una cómoda comunicación entre el cliente `middleware` y el `applet`.
- Se ha desarrollado un sistema el cual puede comercializarse en el futuro, reportando beneficios monetarios al Centro de Identidad y Seguridad Digital y en general a la Universidad de Ciencias Informáticas.

RECOMENDACIONES

Se exponen como recomendaciones para las siguientes fases de la Solución informática:

- Realizar la modelación e implementación de la autenticación asimétrica en la solución para gestionar el control de acceso a instalaciones mediante el uso de tarjetas inteligentes y el `Terminal de Servicio`, para garantizar que tanto el `Applet` de control de acceso, como el terminal de servicio se reconocen como válidos y puedan intercambiar información de forma segura.

- Incorporar mecanismos de autenticación al sistema de control de acceso mediante tecnología MoC[20].

- Continuar con la implementación del estándar ISO/IEC 7816 para soportar diversos mecanismos de almacenamiento y elementos de seguridad.

[20] Match on Card, por sus siglas en inglés.

REFERENCIA BIBLIOGRÁFICA

1. *Sistema de Control de Acceso*. 2010; Available from: http://www.scssa.com.ar/control-de-acceso.htm.
2. *Control de Acceso*. 2011; Available from: http://www.asipro.com.mx/acceso.php.
3. *Sistemas de Control de Acceso*. 2010; Available from: http://www.articulosinformativos.com.mx/Sistemas_de_Control_de_Acceso-a854249.html.
4. Néstor Alonso, S.V. *Control Acceso*. 2008.
5. *Control de Acceso para puertas (control de áreas restringidas)*. 2010; Available from: http://www.elipse.cl/productos/control%20de%20acceso/sistema%20control%20acceso%20fisico%20puerta%20huella%20digital%20tarjetas%20proximidad.html.
6. *Soluciones Integrales de Control de Acceso*. 2011; Available from: http://www.gelb.com.ar/control-de-acceso.html.
7. *Tarjetas Inteligentes* 2003; Available from: http://www.tec-mex.com.mx/promos/bit/bit0103-smart.htm.
8. *Tarjetas inteligentes (smartCards)*. 2007; Available from: http://blog.intelligenia.com/2007/03/tarjetas-inteligentes-smartcards.html.
9. Juan Domingo, S., Ricardo Breito y Juan Carlos. *Tarjetas inteligentes*. 1999 Available from: www.revistasic.com/revista38/pdf_38/SIC_38_bibliografia.PDF
10. Victor Cedeño Moreira. *Smartcard Acces*. 2011; Available from: http://vcanero.tripod.com/smartcards/esp/intro.html.
11. Yurdik, C.M., *BIOMETRÍA EN LAS TARJETAS INTELIGENTES*: Cuba.
12. EBU – TECH. *The Middleware Report*. 2005; Available from: http://tech.ebu.ch/docs/tech/tech3300s.pdf.
13. Bruce, B. *APDU - Application Protocol Data Unit*. 2009; Available from: http://www.birds-eye.net/definition/acronym/?id=1172280403.
14. Jesus Gallaga. *Applet en JavaCard*. 2010; Available from: http://www.javaworld.com/javaworld/jw-07-1999/jw-07-javacard.html?page=1.
15. *Identificación*. 2007; Available from: http://es.thefreedictionary.com/identificaci%C3%B3n.
16. *Autenticación*. 2011; Available from: http://msdn.microsoft.com/es-es/library/syf5yeat.aspx.
17. *Autorización*. 2007; Available from: http://es.thefreedictionary.com/autorizaci%C3%B3n.
18. *Autorización de Sistemas Informáticos* 2008; Available from: http://orientacion.sunat.gob.pe/index.php?option=com_content&view=article&id=396:09-autorizacion-de-sistemas-informaticos-en-la-emision-de-tickets&catid=35:comprobantes-de-pago&Itemid=58.

19. Lucas Manuel. *Sistemas basados en algo poseído: tarjetas inteligentes*. 2008; Available from: http://www.ibiblio.org/pub/linux/docs/LuCaS/Manuales-LuCAS/doc-unixsec/unixsec-html/node112.html.
20. *Arquitectura funcional de una tarjeta inteligente*. 2009; Available from: http://www.gii.upv.es/personal/gbenet/treballs%20cursos%20anteriors-TIM-IIN-INYP-AYPD/smartcards/web/especificaciones.htm.
21. *Introducción en el uso de tarjetas inteligentes*. 2010; Available from: https://zonatic.usatudni.es/es/aprendizaje/aprende-sobre-el-dnie/57-aspectos-tecnicos/204-introduccion-en-el-uso-de-tarjetas-inteligentes.html.
22. GmbH, G.y.D. *Smartcard Aplications*. 2007; Available from: http://www.sumotorrent.com/es/details/581030/Wiley.Smart.Card.Applications.Design.Models.for.Using.and.Programming.Smart.Cards.Jun.2007.pdf.html.
23. Raúl Abad. *Tarjetas Inteligentes y su aplicación*. 2004; Available from: http://www.todomba.com/noticias/marketing/tarjetas-inteligentes-y-su-aplicacion-en-los-programas-de-fidelizacion.html.
24. *Seguridad Tarjetas Inteligentes*. 2009; Available from: http://www.fnmt.es/index.php?cha=companies&scha=19&page=69&spage=145.
25. *GlobalPlatform.* 2003. Version 2.1.1.
26. GlobalPlatform, *Identification cards — Integrated circuit cards*. 2005, Part 4.
27. ISO, *ISO 7816*. 2003/2004.
28. Gemalto, *Estándar PC/SC*. 2011.
29. SIEMENS CORPORATE RESEARCH, I. *RECONOCIMIENTO DE PLACAS DE MATRICULA CON CAMARA INTELIGENTE*. Available from: http://www.invenia.es/oepm:e00307581.
30. *Que son los Sistemas Biométricos*. 2009; Available from: http://eju.tv/2009/04/que-son-los-sistemas-biometricos/.
31. (2011) *Sistema de control de acceso basado en los lectores de control de acceso V-Smart iCLASS y tarjetas inteligentes del Aeropuerto internacional de Ciudad de México (AICM).*
32. (2011) *Registro de asistencia basado en tarjeta inteligente Janus.*
33. *PLATAFORMA DE CONTROL DE ACCESO PARA FUNCIONARIOS Y VISITANTES.* 2010.
34. *Metodologías de desarrollo de software*. 2008; Available from: http://www.rhernando.net/modules/tutorials/doc/ing/met_soft.html.
35. *Metodologías de Desarrollo de Software* 2008; Available from: http://es.scribd.com/doc/2050925/metodologias-de-desarrollo-software.
36. Joskowicz., I.J., *Reglas y Prácticas en eXtreme Programming.* 2005.
37. UML, V.P.f. *UML modeling, requirements capturing and database design*. Available from: http://www.visual-paradigm.com/product/vpuml/provides/.
38. *Runtime Environment Specification*. 2008, Java Card™ Platform.
39. Daniel Perovich, L.R., Martín Varela, *Programación de JavaCards Programación de JavaCards*. 2001.

40. *Microsoft .Net Framework*. 2010; Available from: http://es.kioskea.net/forum/affich-95992-para-que-es-el-net-framework.
41. *El lenguaje de programación C#*. 2011; Available from: http://www.lawebdelprogramador.com/cursos/C_sharp/3869-El_lenguaje_de_programacion_C_.html.
42. *AltovaUmodel*. 2011; Available from: http://www.ramblainf.com/tienda/altova/.
43. *Developer Suite* 2011; Available from: http://developer.gemalto.com/home/dev-tools/developer-suite.html.
44. *Documentación de Visual Studio Team System*. 2011; Available from: http://msdn.microsoft.com/es-es/library/fda2bad5(v=vs.80).aspx.
45. *Arquitectura cliente-servidor*. 2007; Available from: http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html.
46. Joaquin Gracia. *Patrones de diseño*. 2005; Available from: http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php.
47. (2006) *Patrones del "Gang of Four"*.
48. Adriana Gómez, M.d.C.L., Silvina Migani, Alejandra Otazú, *UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE*.
49. Jorge Rodriguez. *Pruebas unitarias*. 2006; Available from: http://blog.continuum.cl/wp-content/uploads/2008/08/pruebas-unitarias.pdf.
50. Jesus Maturana. *Reconocimiento facial para desbloquear tu iPhone, RecognizeMe*. 2011; Available from: http://www.muymac.com/2011/05/18/reconocimiento-facial-para-desbloquear-tu-iphone-recognizeme.
51. *La ciencia y el estudio del calor usando la toma de imágenes térmica infrarroja*. 2009; Available from: http://www.thermology.com/infrarroja.html&usg=__GAuAPiKA-vEwG-HmzHR6uK___lU=&h=219&w=339&sz=19&hl=es&start=3&zoom=1&tbnid=gEDknOlsF1uQgM:&tbnh=77&tbnw=119&ei=cawQTqClHaHx0gH0nv2NDg&prev=/search%3Fq%3DTermograma%2Bdel%2Brostro%26hl%3Des%26lr%3D%26sa%3DN%26tbm%3Disch&itbs=1.
52. *Cerradura con reconocimiento de huellas dactilares*. 2009; Available from: http://www.impresionante.net/28-12-2009/hogar/cerradura-con-reconocimiento-de-huellas-dactilares&usg=__qTBxdXSrJewVwFTqra-D0CYm4PQ=&h=400&w=400&sz=19&hl=es&start=1&zoom=1&tbnid=rldLf3KosgQmVM:&tbnh=124&tbnw=124&ei=Na0QTr7kIs6z0AG6uuihDg&prev=/search%3Fq%3DReconocimiento%2Bde%2Bhuellas%2Bdactilares%26hl%3Des%26lr%3D%26sa%3DN%26tbm%3Disch&itbs=1.
53. *BIOMÉTRICOS.* 2009.
54. *Como mantener en buen estado la Voz* 2008; Available from: Reconocimiento de voz.
55. *Reconocimiento de Firma*. 2011; Available from: http://www.mobbeel.com/es/tecnologia/firma/.

ANEXOS

Anexo 1: Sistemas biométricos.

Figura 8: Reconocimiento de rostro.[50]

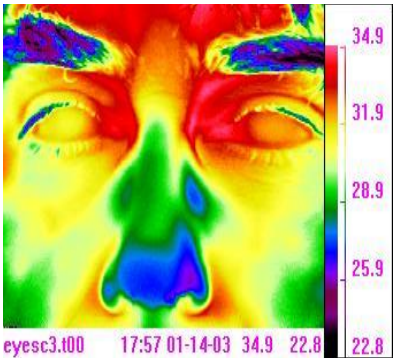

Figura 9: Termograma del rostro.[51]

Figura 10: Reconocimiento de huellas dactilares.[52]

Figura 11: Reconocimiento de geometría de la mano.[53]

Figura 12: Reconocimiento de las venas de las manos.[53]

Figura 13: Reconocimiento de patrones de la retina.[53]

Figura 14: Reconocimiento de voz.[54]

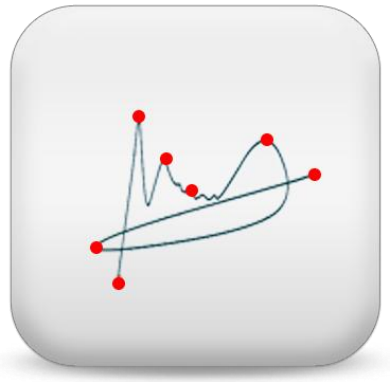

Figura 10. Reconocimiento de firma.[55]

Anexo 2: Fases XP.

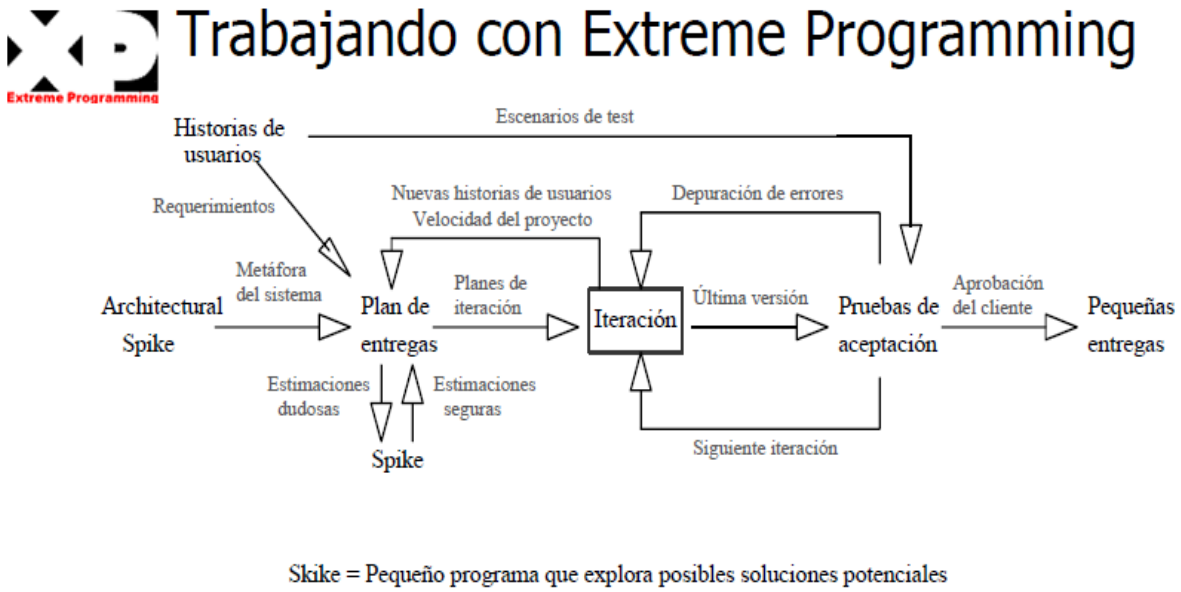

Figura 15: Fases de un proyecto con XP.[36]

Anexo 3: Plan de entregas.

Entregable	Fin Iteración 1	Fin Iteración 2
Sistema Control Acceso	Enero 2011	Mayo 2011

Tabla 11. Plan de entregas.

Anexo 4: Estimación de tiempo de las historias de usuario.

Historia de Usuario	Estimación
Seleccionar lector con el cual se va a establecer la comunicación con la Tarjeta Inteligente.	1
Gestionar comunicación con la tarjeta inteligente.	4
Establecer canal seguro de comunicación con la tarjeta, siguiendo Global Platform.	2
Permitir autenticación del usuario por Número de Identificación Personal (PIN).	3
Cambiar número de identificación personal (PIN).	2
Verificar condiciones de acceso a la información dentro del `Applet` de Control de Acceso.	2
Gestionar datos en la tarjeta	2

Tabla 12.Estimación de tiempo de las historias de usuario.

Anexo 5: Plan de iteraciones.

Iteración	Historia de Usuario	Duración estimada (Semanas)
Iteración 1	Seleccionar lector con el cual se va a establecer la comunicación con la Tarjeta Inteligente.	**9**
	Gestionar comunicación con la tarjeta inteligente.	
	Establecer canal seguro de comunicación con la tarjeta, siguiendo `Global Platform`.	
	Permitir autenticación del usuario por Número de Identificación Personal (PIN).	
Iteración 1	Cambiar número de identificación personal (PIN).	**8**
	Verificar condiciones de acceso a la información dentro del `Applet` de Control de Acceso.	
	Gestionar datos en la tarjeta	

Tabla 13. Plan de iteraciones.

Anexo 6: Estudio de factibilidad.

Nombre	Valor	Justificación
PREC	3.79	Es un proyecto sin muchos precedentes.
FLEX	3.04	Posee cierta flexibilidad de desarrollo (se cuentan con acuerdos de requerimientos preestablecidos)
TEAM	1.12	Interacciones entre los miembros del equipo altamente cooperativas.
RESL	4.28	Se identifican pocos riesgos críticos pero no existen las herramientas para resolverlos.
PMAT	4.68	El nivel de madurez de la organización es utilizadas, resultado de la evaluación según CMM es Nivel 2.
Total (SFj)	**16.91**	

Tabla 14. Factores de escala.

Anexo 7: Estimación de esfuerzo

Nombre	Abreviatura	Valor	Justificación
Fiabilidad del producto y Complejidad.	RCPX	1.30	El sistema presenta un nivel de complejidad alto.
Reusabilidad requerida.	RUSE	1.07	El nivel de reusabilidad es alto.
Dificultad de la aplicación.	PDIF	0.87	Uso de la memoria y almacenamiento bajo, aplicación estable.
Experiencia del personal.	PREX	0.87	El nivel de experiencia en el uso del lenguaje y la aplicación de trabajo es alto.
Aptitud del personal.	PERS	0.83	La capacidad del personal es alta.
Facilidades.	FCIL	0.89	Se utilizan herramientas de modelación que facilitan el trabajo y entornos de desarrollo integrados.
Cronograma de desarrollo requerido.	SCED	1.10	Se utilizó el tiempo planificado para el desarrollo del sistema.
Total (EM)		0.80	

Tabla 15. Factores de escala.

Anexo 8: Puntos de función, entradas y salidas externas.

Nombre de la entrada externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (simple/medio/complejo)
Aplicaciones contenidas dentro de la tarjeta.	1	1	Simple
`Middlewares` externos.	1	1	Simple
Total		2	

Tabla 16. Entradas externas

Nombre de la salida externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (simple/medio/complejo)
Comandos APDU enviados a la tarjeta.	1	1	Simple
Respuestas APDU enviados al `middleware` externo.	1	1	Simple
Total		2	

Tabla 17. Salidas externas

Anexo 9: Líneas de código.

Características	Valor
Puntos de función desajustados	50
Lenguaje (C#)	29
Líneas de código fuente	1450
Miles de líneas de código fuente	1,450 (KSLOC)

Tabla 18. Cálculo de las líneas de código.

Anexo 10: Tarjetas CRC.

SistemaControlAccesoApplet	
Responsabilidades	**Colaboradores**
- Establecer el canal seguro de comunicación con el middleware. - Comprobar que el PIN del usuario sea correcto. - Cambiar el PIN. - Obtener información referente al Usuario. - Almacenar información referente al usuario. - Cerrar el canal seguro de comunicación.	- javacard.framework.* - visa.openplatform.OPSystem

Tabla 20. Tarjeta CRC-1 ¨SistemaControlAccesoApplet¨.

ClienteMiddlewareAppletCA	
Responsabilidades	**Colaboradores**
- Listar lectores conectados. - Establecer comunicación con la tarjeta. - Seleccionar ControlAccesoApplet. - Configurar el canal seguro por Global Platform para el ControlAccesoApplet. - Permitir introducir el PIN. - Permitir verificar el PIN en el ControlAccesoApplet. - Permitir almacenar la información referente al usuario. - Permitir obtener la información referente al usuario.	- Usuario - Util - SmartCard.Client - SmartCard.Core - SmartCard.Core.Common - SmartCard.Core.Utils - SmartCard.GlobalPlatform - SmartCard.GlobalPlatform.Client - SmartCard.Devices.CardReaders - SmartCard.Devices.CardReaders.Win32PCSCWrapper - SmartCard.ISO7816.APDU - SmartCard.ISO7816.APDU.Commands - SmartCard.GlobalPlatform.Security - SmartCard.Securit – MiddlewareAppletCA.Comun – MiddlewareAppletCA.ComandosAPDU – MiddlewareAppletCA.Excepciones - ComandoVerificarPin

	- ComandoCambiarPin - ComandoAlmacenarCarnetIdentidad - ComandoAlmacenarImagenFacial - ComandoAlmacenarNacionalidad - ComandoAlmacenarNivel - ComandoAlmacenarNombreApellidos - ComandoAlmacenarNombreInstitucion - ComandoAlmacenarPais - ComandoCambiarPin - ComandoCrearBufferImagenFacial - ComandoObtenerCarnetIdentidad - ComandoObtenerImagenFacial - ComandoObtenerNacionalidad - ComandoObtenerNivel - ComandoObtenerNombreApellidos - ComandoObtenerNombreInstitucion - ComandoObtenerPais - ComandoObtenerTamannoImagenFacial - ComandoVerificarPin

Tabla 21. Tarjeta CRC-2 ¨ClienteMiddlewareAppletCA¨.

Útil	
Responsabilidades	**Colaboradores**
- Convertir una imagen a un arreglo de byte. - Convertir arreglo de byte a imagen. - Verificar comando APDU de respuesta. - Codificar cadena. - Decodificar cadena. - Convertir de decimal a byte. - Convertir de arreglo de byte a byte. - Convertir de byte a decimal. - Convertir de byte a arreglo de byte.	- SmartCard.ISO7816.APDU

Tabla 22. Tarjeta CRC-3 ¨Útil¨.

Usuario	
Responsabilidades	**Colaboradores**
- Permitir obtener información del usuario. - Permitir cambiar información del usuario.	

Tabla 23. Tarjeta CRC-4 ¨Usuario¨.

Anexo 11: Pruebas Unitarias del `Applet`.

Nombre del elemento	Class	Ins	P1	P2	Data	Le
VerificarPin	90	01	x	x	VerificarPin	x
CambiarPin	90	02	x	x	Datos (Nuevo PIN)	x
AlmacenarCarnetIdentidad	90	03	x	x	Datos (CarnetIdentidad)	x
AlmacenarNombreApellidos	90	04	x	x	Datos (NombreApellidos)	x
AlmacenarNacionalidad	90	05	x	x	Datos (Nacionalidad)	x
AlmacenarPais	90	06	x	x	Datos(País)	x
AlmacenarInstitucion	90	07	x	x	Datos (Institución)	x
ObtenerNombreApellidos	90	08	x	x	x	x
ObtenerCarnetIdentidad	90	09	x	x	x	x
ObtenerNacionalidad	90	10	x	x	x	x
ObtenerPais	90	11	x	x	x	x
ObtenerNombreInstitucion	90	12	x	x	x	x
AlmacenarImagenFacial	90	13	x	x	Datos (ImagenFacial)	x
ObtenerImagenFacial	90	14	x	x	x	Datos (ImagenFacial)
CrearBufferImagenFacial	90	15	x	x	CrearBufferImagenFacial	x
ObtenerTamannoImagenfacial	90	16	x	x	x	0x02
ObtenerNivel	90	17	x	x	x	x
AlmacenarNivel	90	18	x	x	Datos (Nivel)	x

Tabla 24.Según especificaciones de Global Platform.

SW1	SW2	Meaning
'64'	'00'	No specific diagnosis
'67'	'00'	Wrong length in Lc
'68'	'81'	Logical channel not supported or is not active
'69'	'82'	Security status not satisfied
'69'	'85'	Conditions of use not satisfied
'6A'	'86'	Incorrect P1 P2
'6D'	'00'	Invalid instruction
'6E'	'00'	Invalid class

Figura 16: Condiciones de error.

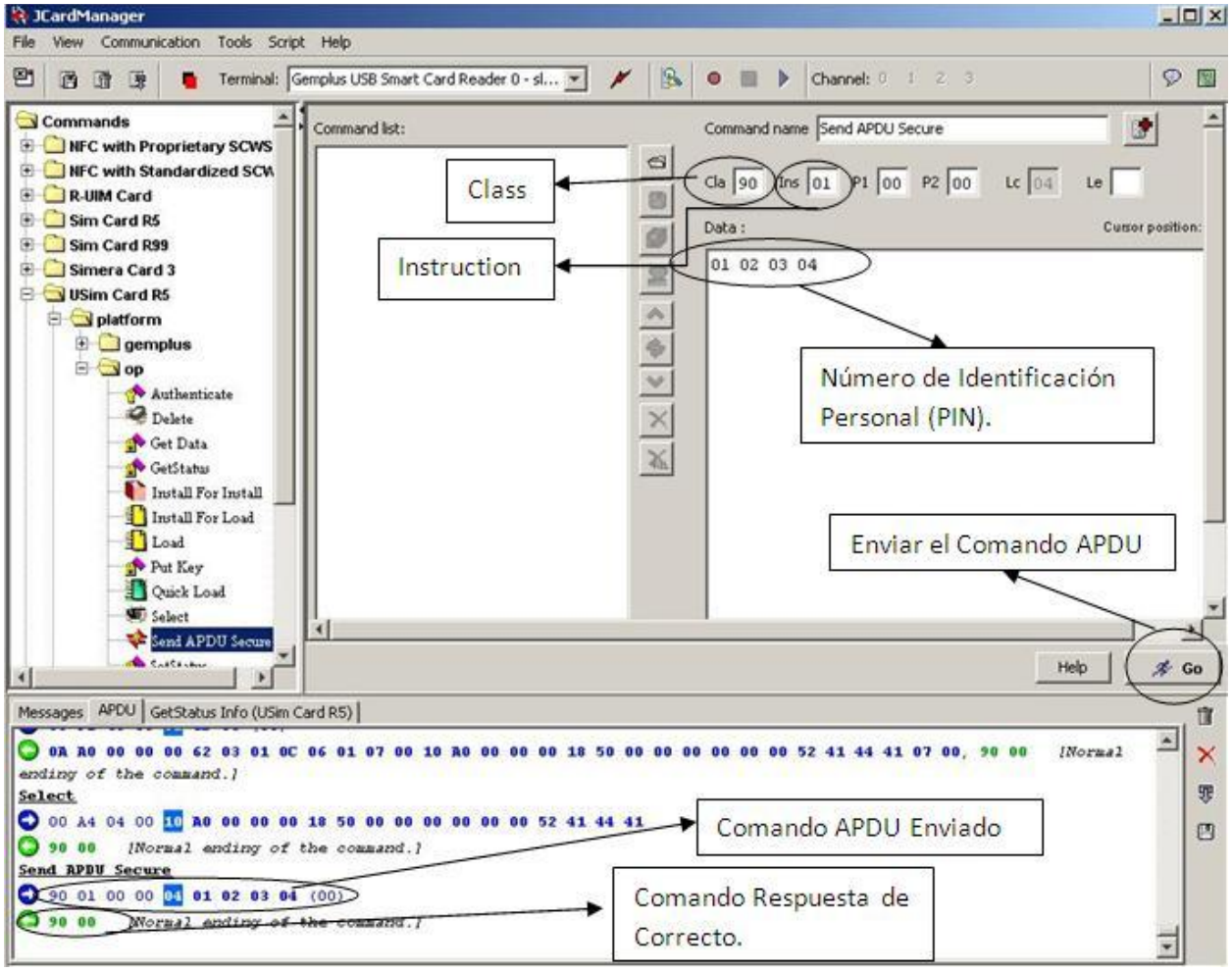

Figura 17: Verificar PIN, caso correcto.

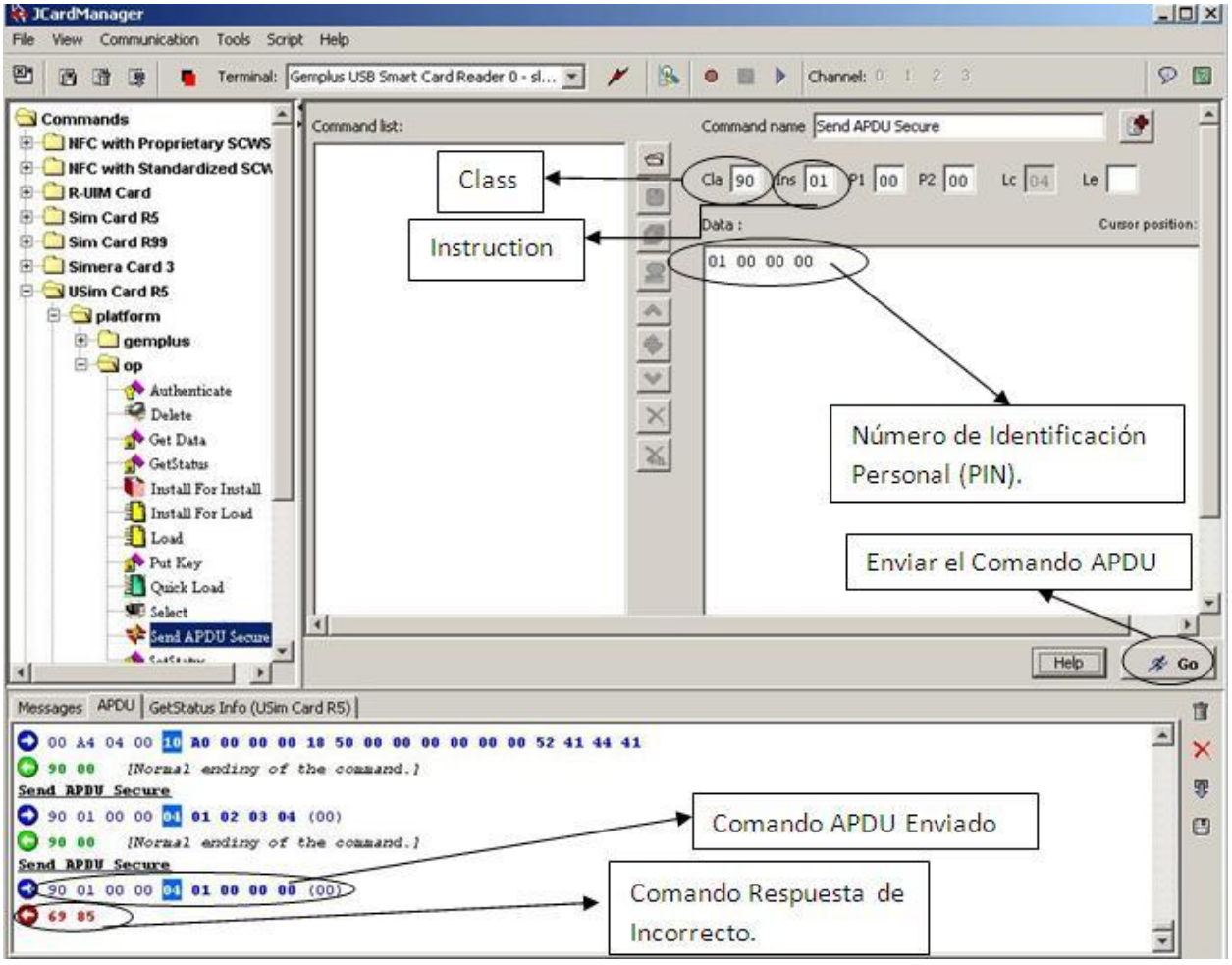

Figura 18: Verificar PIN, caso incorrecto.

Anexo 12: Pruebas Unitarias del `Middleware`.

Prueba de Unidad		
Nombre Prueba: VerificarPinTest		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Última ejecución:**
Ejecutado por: Kirenia García Pérez	**Verificado por:** Orén Fornaris Cabreja	
Descripción: **Para el desarrollo de esta prueba previamente se debe de haber introducido el PIN a verificar, si el mismo es correcto, se autentica satisfactoriamente, de lo contrario se lanza un mensaje de PIN incorrecto.**		
Entrada: Número de Identificación personal(PIN)		
Resultado:		

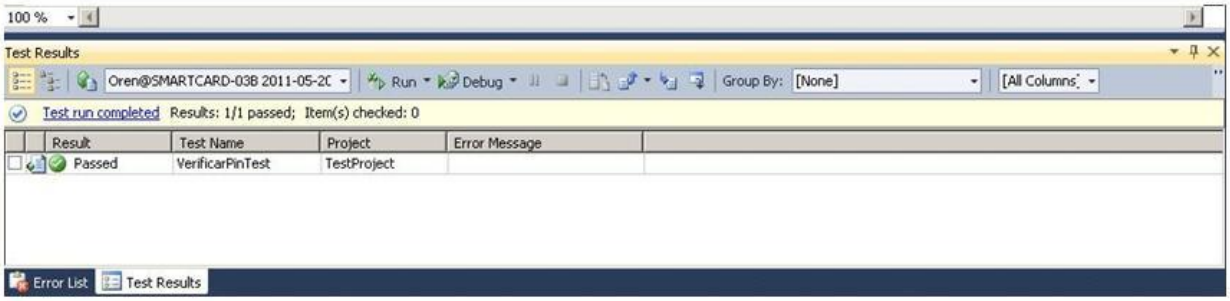

Figura 19: Pruebas de unidad del middleware.

Anexo 13: Pruebas de aceptación.

	Caso de prueba de aceptación
Código de caso de prueba: HU1_CP1	**Nombre de la historia de usuario:** Seleccionar lector con el cual se va a establecer la comunicación con la Tarjeta Inteligente.
Responsable de la prueba: Oren Fornaris Cabreja	
Descripción de la prueba: Prueba de funcionalidad para comprobar la conexión con el lector	
Condiciones de ejecución: Debe existir al menos un lector disponible	
Entrada/Pasos de ejecución: • Seleccionar un lector, en caso que exista más de uno, se escoge el primero o se puede elegir el que se desea.	
Resultado esperado: Queda seleccionado el lector para la comunicación	
Evaluación de la prueba: Prueba Satisfactoria.	

Tabla 25. HU_1CP1 Seleccionar lector con el cual se va a establecer la comunicación con la Tarjeta Inteligente.

	Caso de prueba de aceptación

Código de caso de prueba: HU2_CP2	**Nombre de la historia de usuario:** Gestionar comunicación con la tarjeta inteligente.
Responsable de la prueba: Oren Fornaris Cabreja	
Descripción de la prueba: Prueba de funcionalidad para comprobar la conexión y desconexión con la tarjeta.	
Condiciones de ejecución: Debe existir un lector y una tarjeta, conectados y la conexión de la aplicación tiene que estar abierta.	
Entrada/Pasos de ejecución: • Conectar la tarjeta. • Desconectar la tarjeta.	
Resultado esperado: Queda establecida la conexión con la tarjeta, o cerrada en dependencia del uso requerido, se le muestra un mensaje al usuario.	
Evaluación de la prueba: Prueba Satisfactoria.	

Tabla 26. HU_2CP2 Gestionar comunicación con la tarjeta inteligente.

	Caso de prueba de aceptación
Código de caso de prueba: HU3_CP3	**Nombre de la historia de usuario:** Establecer canal seguro de comunicación con la tarjeta, siguiendo `Global Platform`
Responsable de la prueba: Kirenia Garcia Pérez	
Descripción de la prueba: Prueba de funcionalidad para comprobar el intercambio de información de forma segura entre el `Middleware` y el `Applet` de Control de Acceso.	
Condiciones de ejecución: Debe existir un canal de intercambio de información segura	
Entrada/Pasos de ejecución: • Realizar operaciones en la aplicación.	
Resultado esperado: Queda establecido un canal seguro	
Evaluación de la prueba: Prueba Satisfactoria.	

Tabla 27. HU_3CP3 Establecer canal seguro de comunicación con la tarjeta, siguiendo `Global Platform`.

	Caso de prueba de aceptación
Código de caso de prueba: HU4_CP4	**Nombre de la historia de usuario:** Permitir autenticación del usuario por Número de Identificación Personal (PIN).
Responsable de la prueba: Kirenia Garcia Pérez	
Descripción de la prueba: Prueba de funcionalidad para realizar la autenticación de una persona que porta la tarjeta.	
Condiciones de ejecución: Debe existir un lector y una tarjeta, conectados.	

Entrada/Pasos de ejecución: • Introducir PIN • Dar clic en el botón: Autenticarse
Resultado esperado: Se muestra un mensaje para informar que la autenticación ha sido satisfactoria.
Evaluación de la prueba: Prueba Satisfactoria.

Tabla 28. HU_4CP4 Permitir autenticación del usuario por Número de Identificación Personal (PIN).

	Caso de prueba de aceptación
Código de caso de prueba: HU5_CP5	**Nombre de la historia de usuario:** Cambiar Número de Identificación Personal (PIN).
Responsable de la prueba: Oren Fornaris Cabreja	
Descripción de la prueba: Se introducen los datos requeridos y se da clic en el botón Cambiar PIN para realizar la operación.	
Condiciones de ejecución: Debe existir un lector y una tarjeta, conectados.	
Entrada/Pasos de ejecución: • Introducir PIN anterior • Introducir PIN nuevo • Confirmar PIN • Dar clic en el botón: Cambiar PIN	
Resultado esperado: Se muestra un mensaje para informar que la se realizó el cambio de PIN correctamente.	
Evaluación de la prueba: Prueba Satisfactoria.	

Tabla 29. HU_5CP5 Cambiar Número de Identificación Personal (PIN).

	Caso de prueba de aceptación
Código de caso de prueba: HU6_CP6	**Nombre de la historia de usuario:** Verificar condiciones de acceso a la información dentro del `Applet` de Control de Acceso.
Responsable de la prueba: Kirenia Garcia Pérez	
Descripción de la prueba: Prueba de funcionalidad para verificar el intercambio de APDU entre el lector y la tarjeta.	
Condiciones de ejecución: Debe existir un lector y una tarjeta, conectados.	
Entrada/Pasos de ejecución: • Se realiza una operación en la aplicación.	
Resultado esperado: Se espera respuesta satisfactoria de dicha operación.	
Evaluación de la prueba: Prueba Satisfactoria.	

Tabla 31. HU_6CP6 Verificar condiciones de acceso a la información dentro del `Applet` de Control de Acceso.

Caso de prueba de aceptación	
Código de caso de prueba: HU6_CP7	**Nombre de la historia de usuario:** Gestionar datos en la tarjeta.
Responsable de la prueba: Oren Fornaris Cabreja	
Descripción de la prueba: Prueba de funcionalidad para verificar la gestión de los datos en la tarjeta. : La gestión de datos en la tarjeta está conformada por un primer proceso el cual se encarga de personalizar y leer los datos de la tarjeta, donde obtiene los datos de la persona y los almacena; el otro proceso es el que lee dichos datos y se encarga de visualizarlos en una interfaz.	
Condiciones de ejecución: Debe existir un lector y una tarjeta, conectados, además debe estar autenticado el usuario.	
Entrada/Pasos de ejecución: • Se realiza una operación en la aplicación.	
Resultado esperado: Se espera respuesta satisfactoria de dicha operación.	
Evaluación de la prueba: Prueba Satisfactoria.	

Tabla 32. HU_7CP7 Gestionar datos en la tarjeta