



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS

**PROPUESTA PARA UN CONTROL DE PRÉSTAMOS PARA LAS
BIBLIOTECAS BASADO EN DISPOSITIVOS INTELIGENTES**

ANA LETICIA LEÓN JO

ASESORADA POR: INGENIERO JUAN DANIEL MIJANGOS

GUATEMALA, ABRIL 2005

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**PROPUESTA PARA UN CONTROL DE PRÉSTAMOS PARA LAS
BIBLIOTECAS BASADO EN DISPOSITIVOS INTELIGENTES**

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

ANA LETICIA LEÓN JO

ASESORADA POR: Ing. Juan Daniel Mijangos

AL CONFERÍRSELE EL TÍTULO DE
INGENIERA EN CIENCIAS Y SISTEMAS

GUATEMALA, ABRIL DE 2005

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Sydney Alexander Samuels Milson
VOCAL I	Ing. Murphy Olympo Paiz Recinos
VOCAL II	Lic. Amahán Sánchez Álvarez
VOCAL III	Ing. Julio David Galicia Celada
VOCAL IV	Br. Kenneth Issur Estrada Ruiz
VOCAL V	Br. Elisa Yazminda Vides Leiva
SECRETARIO	Ing. Carlos Humberto Pérez Rodríguez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Sydney Alexander Samuels Milson
EXAMINADORA	Inga. Ligia María Pimentel Castañeda
EXAMINADORA	Inga. Elizabeth Domínguez Alvarado
EXAMINADOR	Ing. Luis Alberto Vettorazzi España
SECRETARIO	Ing. Pedro Antonio Aguilar Polanco

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado

**PROPUESTA PARA UN CONTROL DE PRÉSTAMOS PARA LAS
BIBLIOTECAS BASADO EN DISPOSITIVOS INTELIGENTES**

Tema que me fuera asignado por la Coordinación de la Carrera de Ingeniería en Ciencias y Sistemas con fecha enero de 2003

Ana Leticia León Jo

AGRADECIMIENTOS

A la Universidad de San Carlos de Guatemala

Especialmente a la Facultad de Ingeniería, por ser fuente de conocimientos y gratas experiencias.

Al Ing. Juan Daniel Mijangos

Por su tiempo, conocimientos, paciencia y dedicación al asesorar mi trabajo de graduación.

Al Ing. Arturo Rivera

Por su tiempo, conocimientos, experiencia y disposición que me brindó y que contribuyó a la realización de este trabajo de graduación.

A todas las personas que me brindaron su apoyo y colaboración

Quienes contribuyeron a mi formación como profesional.

ACTO QUE DEDICO

A Dios

Por todas tus bendiciones y por estar siempre a mi lado en todo momento. Gracias Señor por toda tu gracia y misericordia.

A mis padres

Por su inmenso amor y todos sus esfuerzos para lograr mi superación.

A mis hermanos

Por todo su amor y apoyo incondicional.

A mis amigos

Por el cariño, la amistad y el apoyo que me han brindado.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	IV
GLOSARIO	VI
RESUMEN	XI
OBJETIVOS	XII
INTRODUCCIÓN	XIII
1. DEFINICIÓN DE NECESIDADES	1
1.1. Antecedentes	1
1.2. Requerimientos de las bibliotecas	3
1.3. Antecedentes que necesita saber de los usuarios	5
2. DISPOSITIVOS INTELIGENTES	7
2.1. Tarjetas inteligentes (<i>Smart cards</i>)	7
2.1.1. Definición	8
2.1.2. Aplicaciones	11
2.1.3. ¿Por qué una tarjeta inteligente?	13
2.1.4. Tipos de tarjetas inteligentes	16
2.1.5. Estándar ISO 7816	23
2.1.6. Lectores de tarjetas inteligentes	25
2.2. Botones (<i>i-buttons</i>)	29
2.2.1. Definición	29
2.2.2. Aplicaciones	32
2.2.3. Tipos de botones	34

3. ANÁLISIS DE LOS SISTEMAS	37
3.1. Ventajas y desventajas del sistema actual	37
3.1.1. Ventajas	37
3.1.2. Desventajas	37
3.2. Descripción general del sistema inteligente propuesto	38
3.3. Ventajas y desventajas del sistema inteligente propuesto	39
3.3.1. Ventajas	39
3.3.2. Desventajas	40
3.4. Conclusión	41
4. SELECCIÓN DEL DISPOSITIVO ADECUADO	43
4.1. Cuantificación del tamaño necesario para cada dato	43
4.2. Análisis de capacidades de cada dispositivo	49
4.3. Facilidad de uso	49
4.4. Tiempo de vida de los dispositivos	51
4.5. Tiempo de vida de los lectores	53
4.6. Análisis de costos	53
4.7. Conclusión	58
5. DISEÑO DE LA ESTRUCTURA DE DATOS DENTRO DEL DISPOSITIVO	61
5.1. Métodos de seguridad	61
5.1.1. Seguridad de que los datos no sean alterados por terceros	61
5.1.2. Seguridad de que los datos del carné no se pierdan	63
5.1.3. La seguridad de que el dueño del carné sea el que realice el préstamo	65
5.2. Estandarización de la estructura de almacenamiento de datos dentro del dispositivo	66

6. DISEÑO DE LA APLICACIÓN	71
6.1. Descripción general de los sistemas actuales y de lo que sería nuestro sistema inteligente	71
6.2. Flujo de datos	75
6.3. Descripción del flujo de operaciones en nuestro sistema	77
6.4. Compatibilidad con otros sistemas	78
6.5. Módulos a programar	79
6.5.1. Lectura	79
6.5.2. Desencriptar	81
6.5.3. Procesar	82
6.5.4. Encriptar	85
6.5.5. Grabar	86
6.5.6. Volver a hacer tarjeta	86
6.5.7. Otros módulos	87
6.6. Módulos para cada biblioteca	90
6.7. Lenguajes que se pueden usar para programar la aplicación	91
CONCLUSIONES	92
RECOMENDACIONES	95
BIBLIOGRAFÍA	96
APÉNDICE	98

ÍNDICE DE ILUSTRACIONES

FIGURAS

1	Tarjeta inteligente	8
2	Estructura genérica de una tarjeta inteligente	10
3	Tarjeta inteligente de contacto	22
4	Tarjeta inteligente sin contacto	22
5	Tamaños de tarjetas inteligentes	24
6	Dimensión y localización de los contactos	24
7	Lector tipo <i>driver</i>	26
8	Lector con conexión USB	26
9	Lector de tarjetas inteligentes sin contacto	26
10	Lector de tarjetas inteligentes para integrar a cualquier dispositivo electrónico	27
11	Lectoras con escáner de huella digital ACS	27
12	Teclado con lector de tarjeta inteligente y escáner de huellas digitales incorporado	27
13	<i>IButton</i> de 4kbit de EEPROM	29
14	Anillo para botones JAVA	30
15	Llavero para botones	30
16	Lector <i>Blue Dot</i> con conexión al puerto paralelo para botones	31
17	Impresora para tarjetas	55
18	Flujo de datos del sistema inteligente	75
19	Diseño de la base de datos con el mínimo de tablas y campos que debe tener	98

TABLAS

I	Espacio en <i>bytes</i> utilizado por cada tipo de dato	44
II	Valores máximos que se pueden representar con cierta cantidad de <i>bytes</i>	45
III	Espacio estimado para almacenar el encabezado en la tarjeta	46
IV	Espacio estimado para almacenar datos de cada texto en la tarjeta	47
V	Formato de almacenamiento de datos	48
VI	Listado de costos	58
VII	Estructura lógica de los datos dentro de la tarjeta	68
VIII	Ejemplo de estructura de directorios y archivos	69
IX	Estructura de directorios y archivos recomendada utilizando un archivo para cada biblioteca	70

GLOSARIO

Algoritmo	Descripción explícita de la manera cómo se debe de ejecutar alguna tarea en particular (cómo debe ser resuelto un problema).
Autenticación	Verificar la identidad de un usuario u objeto en un computador generalmente, para acceder a recursos de un sistema.
APDU's	<i>(Application Protocol Data Unit)</i> Protocolo que establece la estructura del paquete de datos que contiene una instrucción completa o una respuesta completa desde la tarjeta.
API	<i>(Application Programmatic Interface)</i> , es una definición de cómo se comunican entre sí dos entidades de software, o una de software con una de hardware. En el caso de las tarjetas, estos sirven para que un programa en la PC se pueda comunicar con una tarjeta. El más común hoy día es el PC/SC.
Biometría	Técnica que estudia las características físicas de una persona como la huella digital, la estructura de los ojos o el patrón de voz, por medio de métodos estadísticos y el cálculo.
CDS-ISIS (comúnmente denominado MicroIsis)	<i>(Computarized Documentation System - Integrated Set for Information System)</i> . Sistema de almacenamiento y recuperación de información, diseñado para el registro de documentos, permite construir y administrar bases de datos estructuradas y su uso está dirigido principalmente al área bibliográfica.

Criptograma	Mensaje o texto cifrado.
COS	<i>(Card operating system)</i> Sistema operativo de las tarjetas inteligentes con microprocesador. Es el <i>software</i> que controla la operación de las tarjetas, maneja las aplicaciones y provee servicios tales como seguridad, manejo de memoria, acceso a primitivas de lectura y escritura y de criptografía.
Desencriptar	Proceso por medio del cual, se decodifica o descifra la información.
DLL	Un fichero DLL es un fichero que almacena funciones ejecutables o datos que pueden ser usados en una aplicación Windows.
<i>Drivers</i> o Manejadores	Conjunto de instrucciones que permiten que el sistema operativo de una máquina se comunique con un dispositivo periférico sin la necesidad de conocer cómo funciona este.
EEPROM	<i>(Electrically Erased Programmable Read Only Memory)</i> Memoria de sólo lectura programable borrable eléctricamente. Es una memoria no volátil, es decir, conserva su contenido una vez que se ha apagado la fuente de alimentación eléctrica, el usuario puede programarla y puede ser rescrita varias veces.
Encriptar	Proceso por medio del cual se codifica o cifra una información por medio de un algoritmo utilizando una clave, con el fin de evitar que terceras personas tengan acceso a ella.

Epóxico	Sustancia química que polimerizada se usa como plástico para estructuras, revestimientos y adhesivos.
EPROM	(<i>Erasable Programmable Read Only Memory</i>) Memoria programable de sólo lectura, que puede ser borrada.
Formato binario	La información en formato binario, se encuentra expresada en términos de ceros (0) y unos (1), donde cada cero o uno, ocupan 1 <i>bit</i> de espacio.
GSM	Estándar para teléfonos digitales móviles, describe especificaciones técnicas.
ISO	(<i>International Standards Organization</i>) Organización internacional de reglas o normas. Organización encargada de crear estándares para diversos productos y servicios.
Java	Lenguaje de programación utilizado para crear aplicaciones de páginas del Internet.
Mapa de la tarjeta	Estructura de almacenamiento de los datos dentro de la tarjeta.
Microchip	A veces llamado <i>chip</i> , es un circuito integrado fabricado sobre materiales como el silicio, a una escala microscópica, para llevar la lógica de programas (microprocesadores) o para funcionar como memoria (<i>chips</i> de memoria o <i>chips</i> de RAM).

MFN	<i>(Master File Number)</i> Número único, que es asignado de manera automática y secuencial por CDS/ISIS a cada registro que se ingresa a una de sus bases de datos y representan el orden cronológico de entrada del registro.
NV RAM	<i>(Non Volatile Random Access Memory)</i> Memoria de acceso aleatorio no volátil.
Outsourcing	Sistema utilizado por empresas grandes, que rentan los servicios de compañías chicas para efectuar proyectos pequeños en lugar de efectuarlas ellas.
PC/SC	<i>(Personal Computer/Smart Card)</i> . Grupo de trabajo que desarrolla especificaciones para solucionar los problemas de interoperabilidad de las tarjetas inteligentes.
Personalización	Proceso en el cual se cargan datos, como las aplicaciones y la información personal de una persona, en la tarjeta.
PIN	Número de identificación personal, que consiste en un número o código que el dueño debe teclear para asegurar que él es el verdadero dueño de la tarjeta.
Protocolo	Son especificaciones que describen las reglas y procedimientos que se utilizan para la transmisión de datos. Si dos productos de diferente proveedor no utilizan el mismo protocolo, no podrán comunicarse entre ellos.
PVC	<i>(Polyvinyl Chloride)</i> Material de polivinilo.

RAM	<i>(Random Access Memory)</i> Memoria de acceso aleatorio, es una memoria volátil y se emplea para almacenar datos temporales.
ROM	<i>(Read Only Memory)</i> Memoria de sólo lectura. La información que almacena no puede ser alterada. En las tarjetas inteligentes con microprocesador, contiene el sistema operativo de la tarjeta.
SIM	Módulo de Identificación del Suscriptor. Método de identificación y seguridad en teléfonos móviles, que permite que un usuario se identifique y use los servicios de la red GSM independientemente del móvil usado.
Sistema de clasificación decimal de Dewey (CDU)	Sistema de clasificación de textos utilizado por las bibliotecas, que parte de la división de las ciencias en nueve clases principales como filosofía, religión y ciencias naturales entre otros, se le asigna a cada disciplina un número decimal y cada nueva cifra que se añade es una subdivisión de la anterior.
Sistema operativo	Conjunto de programas que permiten el uso del <i>hardware</i> de una computadora, se encarga de administrar sus recursos y de planificar los procesos que utilizan los mismos.
Sustrato	Soporte que permite la adherencia entre la capa sensible a la luz y el vidrio o las materias plásticas (lámina sobre el que se efectúa la deposición de películas delgadas), en la fabricación de circuitos integrados.
Winisis	Programa que es una versión de CDS/ISIS, para almacenar los datos de los textos que se tienen disponibles en una biblioteca.

RESUMEN

Las tarjetas inteligentes y los botones son dispositivos muy utilizados para aplicaciones como el comercio electrónico y el acceso a instalaciones, áreas restringidas o a sistemas basados en computadoras, pueden ser usados como monedero electrónico, control de parqueo, ficha médica, etc. En este caso, se aplica al control de préstamo de textos de las bibliotecas.

La idea fundamental, es que cada usuario cuente con su carné de biblioteca siempre, al evitar que se le retenga su carné, y que cualquier persona autorizada pueda conocer, en cualquier momento, si el usuario está inscrito, a qué facultad y carrera pertenece, conocer sus datos generales y su solvencia con las bibliotecas para tomar cualquier decisión, la que es importante saber si el usuario tiene mora o no en alguna biblioteca.

Para determinar qué dispositivo es el indicado, es necesario analizar las características de los mismos, determinar los requerimientos básicos del sistema y diseñar la estructura de la misma, de acuerdo a las necesidades a satisfacer, que en este caso es, mantener un mejor control de los préstamos de las bibliotecas.

OBJETIVOS

- **General**

Diseñar un medio de identificación único para todas las bibliotecas, que sea fácil de usar, confiable y que brinde información actualizada en el momento que se use, permitiendo así, un mejor control sobre la cantidad de textos prestados a cada usuario.

- **Específicos**

1. Identificar los requerimientos de los bibliotecarios.
2. Presentar los distintos medios de almacenamiento que existen en la actualidad, con capacidad de almacenar información actualizada de las personas.
3. Analizar el estado del sistema de identificación universitaria actual y los resultados que se obtendrían con un sistema de identificación inteligente.
4. Presentar el dispositivo de almacenamiento que más se adapte a las necesidades de los administradores de las bibliotecas.
5. Diseñar una estructura de almacenamiento adecuada, funcional, segura, rápida, y confiable, que pueda usarse dentro del dispositivo inteligente.
6. Diseñar la estructura de la aplicación que manipule la información sobre la que se desee llevar un control, de una manera segura, rápida y estructurada.

INTRODUCCIÓN

Las tarjetas inteligentes y los botones son tecnologías que actualmente se están utilizando en todo el mundo para diversas aplicaciones. Entre las funciones o aplicaciones para las cuales las podemos utilizar tenemos que, en ellos podemos almacenar datos actualizados sobre las personas u objetos sobre los cuales se desea llevar un control, contraseñas, claves de acceso o características de las personas como su huella digital o el patrón de su voz, que los hacen útiles para el comercio electrónico y el acceso a instalaciones, áreas restringidas o a sistemas basados en computadoras.

Ciertas versiones de tarjetas inteligentes ofrecen la funcionalidad de manejar varias aplicaciones en una sola tarjeta, lo que permite usarlos como monedero electrónico, control de parqueo, ficha médica, etc. en un solo dispositivo, y tienen características que lo hacen seguro contra cualquier intento de lectura no autorizada.

Los botones a su vez, tienen versiones que permiten tomar la temperatura o llevar el control de fechas y horas de vencimiento, que los hacen útiles para el control de productos perecederos, y versiones basados en la tecnología de JAVA que permiten su uso para aplicaciones que requieren alta seguridad en los datos almacenados.

Para mostrar algunos de los beneficios que tienen estos dispositivos, se ejemplifica su uso, proponiendo un sistema para el control de préstamos en las bibliotecas de la Facultad de Ingeniería, mostrando las necesidades que se desean satisfacer con un sistema basado en dispositivos inteligentes y el proceso de selección del dispositivo más adecuado, de acuerdo a las necesidades y los costos para implementarlo. Se propone un esquema del diseño de la estructura que se tendrá dentro del dispositivo escogido y un diseño del flujo de la información dentro del sistema.

1. DEFINICIÓN DE NECESIDADES

1.1. Antecedentes

Las bibliotecas son parte esencial de los centros educativos debido a que proporcionan al estudiante el material de apoyo que necesitan para realizar sus tareas e investigaciones. Para que todos los estudiantes tengan acceso a este servicio, los bibliotecarios procuran mantener copias disponibles de todos los textos, lo que en muchas ocasiones es muy difícil, debido a que se cuenta con pocas unidades de cada texto, por lo que los encargados de las bibliotecas han optado por no admitir el préstamo externo de textos (es decir, que los usuarios no saquen los textos de la biblioteca para llevárselos a sus casas).

Algunas bibliotecas, no realizan préstamos externos a alumnos regulares, pero sí proporcionan este servicio a catedráticos, estudiantes de maestrías o de especialización (como los que estudian Licenciatura en Física), estudiantes graduandos o que se encuentran realizando su trabajo de graduación. El proyecto Biblioteca USAC-UTRECHT, por ejemplo, realiza de manera restringida los préstamos, y exige que presenten una carta de un licenciado que se haga responsable del préstamo externo al estudiante que lo solicita.

El tiempo que se les da a los usuarios para que permanezcan con los textos prestados, varía de biblioteca en biblioteca y depende del tipo de texto (tesis o libro normal) y el tipo de usuario que realiza el préstamo. Un estudiante que realiza su tesis, por ejemplo, tiene la oportunidad de prestar 2 tesis en la Biblioteca MCC y llevárselos a su casa, durante 15 días.

Para aquellas personas que deseen sacar los textos de las bibliotecas sin salir de la universidad, deben cumplir con ciertas condiciones que varían de biblioteca en biblioteca. Las bibliotecas USAC-UTRECHT, CICON y ERIS, por ejemplo, exigen que además de presentar su carné, dejen todas sus pertenencias para asegurar que regresen con los textos y no se los lleven fuera de la universidad, mientras que la Biblioteca MCC, es más flexible y sólo pide el carné del estudiante.

Antiguamente, cuando aún no existían los carnés plásticos con código de barras que se usan actualmente, los estudiantes habían tenido problemas en el momento de prestar textos, al no contar con su carné de la universidad por haberlo dejado en la Biblioteca Central. Para solucionar este problema, los encargados de las bibliotecas de la Facultad de Ingeniería, admitieron el préstamo de textos con cualquier documento de identificación.

En aquellas bibliotecas con políticas restringidas, no surgieron mayores problemas al admitir préstamos con cualquier documento de identificación, sin embargo, en la Biblioteca MCC sí, pues se observó que los estudiantes terminaban prestando más textos de los que estaban permitidos, al presentar cualquier identificación cada vez que los solicitaban. Por esta razón, esta biblioteca instituyó un carné exclusivo para préstamos en ella, sin el cual, no les era permitido que los estudiantes realizaran préstamos.

Un problema que surgió a partir de la creación de este carné exclusivo, fue el de la reposición de carnés. Para reponerlos, los encargados de realizar esta labor, debían investigar los antecedentes de los estudiantes para verificar que los datos que proporcionaban eran íntegros, revisando en sus grandes archivos físicos, que no tuvieran textos sin devolver y luego teclear con una máquina de escribir los datos en el nuevo carné, volviendo esto, un trabajo tedioso y que absorbía mucho tiempo de trabajo de ellos, debido a la gran cantidad de carnés que debían entregar diariamente.

Actualmente, se tiene el carné plástico para los estudiantes de Ingeniería, con el que se tiene la posibilidad de prestar textos en la Biblioteca Central sin que estos les retengan el carné, con este beneficio, la Biblioteca MCC optó por pedir el carné plástico de Ingeniería y eliminar el carné que se utilizaba, con el fin de reducir la carga de trabajo de la persona responsable de la emisión de los carnés.

Además, la Biblioteca MCC mantiene un listado de estudiantes morosos, listado que les toma tiempo considerable para obtenerlo (al revisar sus archivos físicos manualmente) y no se actualiza con regularidad. Este listado lo utilizan para verificar si el usuario tiene algún texto no devuelto, si el usuario no tiene prestado ningún texto, le pegan al carné plástico una calcomanía para indicar que el estudiante está solvente en esta biblioteca y evitar tener que revisar el listado la próxima vez que vaya a prestar un texto. Con este método, surge el problema de que el usuario logre conseguir las mismas calcomanías y se lo pegue al carné, haciendo parecer que está solvente.

Otro problema que ha surgido a raíz del nuevo carné, es que hasta el momento, los encargados no saben cómo van a verificar el próximo año (2005) si el estudiante está inscrito o no, puesto que el carné actual tiene vigencia solamente en este año (2004), y si el usuario no cuenta con un carné válido para en el 2005, no podrá prestar textos.

Además de estos problemas, los encargados de las bibliotecas se han visto obligados a pagar el costo de los textos que nunca fueron devueltos debido a que no pudieron localizar a las personas a quienes les fueron prestados.

1.2. Requerimientos de las bibliotecas

Se realizaron entrevistas a los distintos encargados de las bibliotecas de la Facultad de Ingeniería, con el propósito de conocer el procedimiento utilizado para realizar los préstamos, los problemas y necesidades que tienen.

El sistema de préstamos en cada biblioteca, varía dependiendo de sus políticas de préstamo y del grado de especialización del contenido de los textos que tienen a disposición de todos, por lo que también varían los tipos de personas que llegan a consultar los textos, es decir, desde estudiantes normales hasta estudiantes de maestrías o catedráticos.

Los encargados de las bibliotecas entrevistados, presentaron los siguientes requerimientos:

- Evitar que se den los casos en el que el usuario realiza préstamos con el documento de identificación de otra persona.
- Contar con la firma del usuario en el carné, como parte de los datos a verificar para prestar textos.
- Facilitarles el trabajo de verificar los antecedentes de los usuarios, antes de prestarles los textos que solicitan.
- Poder controlar los préstamos a aquellas personas que no estén inscritas en la universidad, al saber de manera concreta y verificable, si están inscritos o no.
- Tener la capacidad de conocer quiénes no han devuelto textos, pasándose de la fecha límite que se les dió, para utilizar los textos prestados, y contar con sus datos para realizar reclamos, sin necesidad de hacer esto a mano.
- Llevar el control de las personas encargadas de las bibliotecas, para saber quién es el responsable del préstamo de un texto a cualquier usuario, es decir, quién realizó la operación de préstamo y verificó los datos del usuario responsabilizándose así, de tal préstamo y del pago del costo del mismo, en caso de que no lleguen a devolver los textos.

1.3. Antecedentes que necesita saber de los usuarios

Antes de prestarle un texto a cualquier usuario, es necesario obtener cierta información de ellos y que cumplan con ciertas condiciones, como un medio para prevenir la pérdida de los textos por la falta de devolución de los mismos. Los requisitos que piden los encargados de las bibliotecas, para cualquier solicitud de préstamo externo, son los siguientes:

- Estar inscrito en el año en curso, en la Universidad de San Carlos de Guatemala
- Saber si pertenecen a la facultad al que pertenece la biblioteca
- Conocer los datos personales del estudiante (nombre completo, dirección, teléfono)
- Presentar su carné
- Que no tenga más de 2 ó 3 textos (dependiendo de cada biblioteca) prestados
- Saber la fecha y hora en el que se les prestó los textos y/o la fecha en que deben devolverlos

Además de estos requisitos, se deben de cumplir aquellos que son especiales de cada biblioteca, en la Biblioteca USAC-UTRECHT, por ejemplo, necesitan la carta de autorización de préstamo, firmada por un licenciado (catedrático) que se responsabilice por el mismo; y en la Biblioteca MMC, tener la fotocopia del certificado de cursos aprobados o la matrícula consolidada, para préstamo de tesis por 15 días.

2. DISPOSITIVOS INTELIGENTES

2.1. Tarjetas inteligentes (*Smart Cards*)

A medida que avanza los años, la tecnología ha ido evolucionando en busca de diversas formas para facilitar las tareas cotidianas y de mantener un estricto control sobre los datos confidenciales de toda persona o empresa. El uso de tarjetas inteligentes se ha popularizado principalmente en Europa y Asia, no así en Estados Unidos. La principal aplicación que ha tenido es la telefonía, en donde se estima que en 1996 se vendieron 420 millones de tarjetas telefónicas. Para julio del año 1997, más de 90 instituciones bancarias en el mundo tenían funcionando sistemas basados en tarjetas inteligentes, y ya existían los llamados ATM (auto cajeros) personales, por medio de los cuales se puede acceder vía telefónica, a las computadoras de un banco y trasladar dinero de su cuenta a la tarjeta.

En Francia, las tarjetas de monedero electrónico llamadas Moneo, fueron introducidas en el 2001 en varias regiones y el servicio llegó a París. Para finales del 2003, unos 850,000 consumidores utilizaban tarjetas Moneo en forma regular en 80,000 tiendas de comestibles, estacionamientos o máquinas expendedoras automáticas, según informe de la empresa tecnológica BMS que lanzó el proyecto.

Después de ver la efectividad y experiencia que Europa tuvo en el uso de tarjetas con *chip*, en Monterrey, México, las empresas Santander Serfin y Banamex pusieron en marcha un proyecto piloto en septiembre del 2002, lanzando sus productos UniSantanderK y B-Smart, con el fin de ver cuál era el impacto hacia el consumidor final y hacia los comercios, pero de ser un programa piloto se convirtió rápidamente en una migración de tarjetas de banda magnética a tarjetas con *chips*.

2.1.1. Definición

Una tarjeta inteligente es una tarjeta de plástico, generalmente fabricada de PVC, del tamaño de una tarjeta de crédito (85mm x 45mm), que tiene incrustado en él, uno o más *chips* y funciona como un medio de almacenamiento de información, que puede contar con o sin un microprocesador que ejecute un sistema operativo de tarjeta (COS). Se les llama inteligentes ya que, por medio de su procesador, tienen la capacidad de administrar la información que contienen: leer, escribir, administrar la seguridad, usar medios criptográficos para asegurar las claves, etc.

Figura 1. Tarjeta inteligente



Fuente: [www1.slb.com/Schlumberger Training SIM cards.htm](http://www1.slb.com/Schlumberger%20Training%20SIM%20cards.htm)

Estas tarjetas son utilizadas para transportar información de un sistema a otro sin necesidad de que estos sistemas se encuentren funcionando en línea, además de proveer seguridad contra accesos no autorizados a la información. Con la información que esté almacenada en las tarjetas, se pueden realizar diversas tareas dependiendo de la aplicación que se esté utilizando para procesar dicha información.

Su predecesor, la tarjeta con banda magnética convencional, se desarrolló a finales de los años 60 para distintas aplicaciones. Entre estas aplicaciones podemos mencionar la de la banca, que han permitido a los clientes de los bancos y entidades de ahorro, tener acceso de forma rápida, efectiva y segura a su cuenta, a través de cajeros automáticos y sistemas de punto de venta. Además, se utilizó para facilitar la facturación y embarque de pasajeros en las líneas aéreas.

Con el pasar de los años, se han utilizado muchas tarjetas con banda magnética para distintos fines financieros, sin embargo, esta presenta la desventaja de que transporta poca información, no ofrece fiabilidad (se puede dañar muy fácilmente), ni seguridad en la información que lleva (la información puede ser leída por una gran diversidad de dispositivos, sin necesidad de códigos o permisos especiales).

La tarjeta inteligente surge entonces, de la evolución de la tarjeta de plástico convencional y su combinación con un circuito integrado (*chip*), proporcionando portabilidad, seguridad sobre los datos y conveniencia para su transporte. Bull CP8, Schlumberger y Philips fueron, a principios de los 80, los tres primeros fabricantes de tarjetas inteligentes. Es a principios de los 90 cuando las tarjetas inteligentes inician su despegue al empezar la telefonía móvil GSM, empleando tarjetas con de memoria.

Con el fin de ofrecer más servicios, mayor seguridad y más sencillez, MasterCard, Visa y American Express comenzaron a incorporar el *chip* en todas las tarjetas de crédito que se emitieron a partir de finales de 2002. La tecnología utilizada ha sido desarrollada por Europay, Mastercard y Visa (EMV), quienes crearon un estándar para todas las tarjetas.

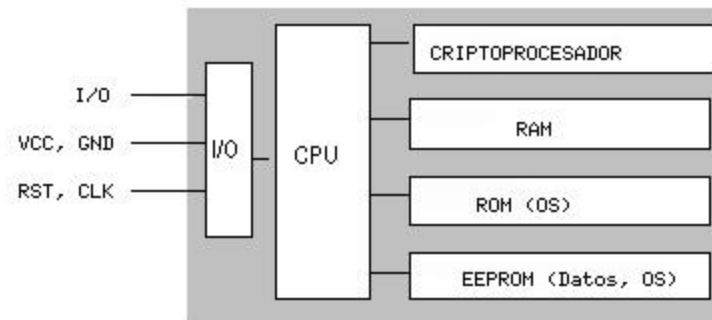
Las tarjetas inteligentes ofrecen varias ventajas y diversas posibilidades, entre las cuales podemos mencionar:

- Almacenamiento de datos como certificados digitales, credenciales y claves de acceso
- Proteger el proceso de conexión y autenticación de usuarios a computadoras y redes
- Permitir el comercio electrónico seguro
- Protección de información importante
- Cifrado de información, que lo hace útil para sistemas de autenticación de personas.

- Permite la utilización de una única tarjeta para varias aplicaciones distintas
- Generar menores costes por transacción que las tarjetas de plástico convencionales al no tener la necesidad de estar en línea para realizar consultas.
- El coste por tarjeta también se ve reducido, ya que tiene mayor tiempo de vida y esta puede actualizarse.

Las tarjetas inteligentes pueden contener un *chip* con una unidad de procesamiento central y varios espacios de memoria a largo y corto plazo. Algunas versiones tienen incorporado un circuito especial de coprocesamiento para operaciones criptográficas que permiten acelerar el proceso de codificación y decodificación de mensajes o la generación de señales digitales para validar la información transferida.

Figura 2. Estructura genérica de una tarjeta inteligente



Fuente: <http://es.tldp.org/Manuales-LuCAS/doc-unixsec/unixsec-html/node112.html>

Entre los fabricantes del *chip* de silicio que se inserta en las tarjetas, están Motorola, Siemens y SGS-Thompson. Empresas como Gemplus, Axalto (antes SchlumbergerSema) y Oberthur, entre otros, insertan el *chip* en las tarjetas configurándolo con su propio sistema operativo (COS), pero el fabricante más grande de tarjetas es Gemplus. En cuanto a terminales, Bull es una de las empresas que se especializa en ellos.

Netscape y Microsoft buscan introducir las tarjetas inteligentes en el campo del Internet, como un medio para el acceso seguro al mismo. También MUSCLE, el movimiento para el uso de tarjetas inteligentes en ambiente Linux, trabaja en un proyecto para coordinar el desarrollo de tarjetas inteligentes, aplicaciones y lectores capaces de trabajar en un ambiente Linux.

2.1.2. Aplicaciones

En teoría, la misma tarjeta podría servir como identificación personal, tarjeta de crédito, tarjeta de cajero automático, tarjeta de teléfono, boleto de pasaje, almacenador de información médica y sustituto de dinero en efectivo para transacciones en persona o en el Internet. Para los proveedores de servicios, esta versatilidad se traduce en importantes ahorros en recursos, comunicaciones y en la disminución del fraude. Los usos que se le pueden dar son limitados únicamente por la imaginación de las entidades emisoras y por la aceptación de los consumidores.

En los Estados Unidos, las personas fueron motivadas a utilizar las tarjetas inteligentes debido a que pueden ser utilizadas para múltiples tareas, entre las cuales podemos mencionar:

- Establecer nuestra identidad al ingresar en un proveedor de acceso a Internet o a un banco en línea, sin la necesidad de recordar los números de cuenta.
- Pagar el parqueo, el viaje en trenes, metros subterráneos o buses, o bien, comprar gasolina, sin utilizar papel moneda.
- Proporcionar instantáneamente, información personal a hospitales o doctores, sin la necesidad de llenar formularios.
- Realizar compras en tiendas electrónicas a través de Internet con mayor seguridad, ya que la cantidad a pagar se debita del saldo que está dentro de la tarjeta, no de una cuenta bancaria o de crédito.

En la actualidad, las tarjetas inteligentes son utilizadas para los siguientes servicios:

- Tarjetas de telefonía móvil, que permiten tener un registro del abonado y una clave de acceso.
- Tarjetas de crédito o débito, sustituyendo a las tarjetas de banda magnética y trabajando en un entorno más seguro.
- Tarjetas de salud, que puede contener información identificativa y un historial clínico o información sobre alergias y enfermedades crónicas, así como también, información de los tratamientos que esté llevando el paciente.
- Monedero electrónico, en donde el *chip* contiene información acerca del saldo monetario de las personas, la cual, se carga en cajeros automáticos. Permite sustituir a las monedas en las pequeñas compras realizadas en máquinas expendedoras, quioscos, parquímetros, teléfonos públicos, etc.
- Tarjetas telefónicas, en este sector han tenido un mayor uso. El *chip* contiene información acerca del saldo pendiente de uso y su lectura se realiza en las cabinas telefónicas preparadas para ello.
- Servicios comunes en universidades y tarjetas de pago de TV. Varias universidades españolas han puesto en marcha proyectos basados en tarjetas inteligentes, donde cada estudiante posee una tarjeta que los identifica y les permite tener acceso a todos los servicios de la universidad (como la biblioteca, fotocopias, etc.), permitiendo que también se use como tarjeta de crédito y monedero electrónico.
- Protección de acceso al PC (no permite el acceso al ordenador sin ella), bloqueo del PC en caso de retirar la tarjeta y desbloqueo por el usuario, protección por huella digital.
- Protección de información y el uso del *software* de la empresa, una vez dentro de la aplicación cada tarjeta permite el acceso a unos determinados datos.
- Control de acceso y de presencia, limitando y controlando el acceso a áreas restringidas, edificios, oficinas, clubes, administración, ordenadores, etc.

- Control de navegación en Internet por medio de un programa desarrollado para la gestión de cybercafés, cyberbibliotecas, cyberaulas y cybersalas, donde la navegación virtual o el uso de *software* esté sujeto a cobro o tiempo limitado.

Resumiendo, se han implementado muchas soluciones basadas en tarjetas inteligentes para simplificar aplicaciones como las comunicaciones públicas e inalámbricas, transacciones financieras, transporte, educación, atención médica, identidad, control de acceso físico, comercio electrónico, seguridad en Internet, control de acceso lógico e informática.

2.1.3. ¿Por qué una tarjeta inteligente?

Las tarjetas inteligentes ofrecen muchas ventajas que explican el gran éxito que han tenido. Entre las ventajas más importantes, están:

Primero, su capacidad de almacenamiento, pues pueden transportar información de diez a cien veces más que las tarjetas de banda magnética, almacenándola de manera más confiable y segura. Además, es posible guardar información diversa, al habilitar el transporte de credenciales y otra información privada entre equipos, tanto en la oficina como en el hogar o en los viajes.

Segundo, la complejidad de las operaciones que puede realizar, pues pueden ejecutar tareas complejas junto con una terminal, como la verificación de la validez de la información almacenada en la tarjeta por medio de varias preguntas y respuestas entre la tarjeta y la terminal de lectura.

Tercero, su conveniencia, ya que son tarjetas reescribibles, permiten que los datos almacenados puedan ser actualizados; se pueden utilizar para múltiples aplicaciones variadas, y hace de las transacciones, operaciones más rápidas.

Cuarto, es económico, debido a que:

- Producir e implementar las tarjetas resulta barato, pues cualquier terminal operadora de monedas, como se puede constatar en el servicio telefónico en Guatemala, puede ser reemplazado por una terminal basada en tarjetas, reduciendo el riesgo de fraude.
- Se reduce el costo por tarjeta, puesto que tiene mayor tiempo de vida, es capaz de actualizarse y se puede utilizar una tarjeta para varias aplicaciones.

Quinto, proporciona mayor grado de seguridad, ya que:

- Las tarjetas y los lectores utilizan métodos de autenticidad para identificarse entre ellos.
- El *chip* incrustado no puede cambiarse.
- Para proteger la información existe un control de acceso por medio de distintos sistemas, ya sea mediante un código PIN para identificar al usuario a través de una contraseña dentro de otra tarjeta (autenticación del portador) o mediante el uso de una terminal que contenga claves secretas para la autenticidad de las tarjetas (autenticación a través de un sistema de claves diversificadas).
- El *chip* es a prueba de falsificación y no puede ser duplicado.
- Capacidad de detección de ataques por rayos X y luz ultravioleta, voltajes inusuales y cambios de frecuencia de reloj.
- La información contenida en la tarjeta, no puede ser leída por cualquier aparato lector (por el mismo proceso de autenticación entre tarjeta y lector por medio del uso de claves cifradas), a diferencia de las tarjetas de banda magnética que pueden ser leídos y pueden ser manipulados por personas con conocimiento y medios adecuados con relativa facilidad.

Sexto, reduce la necesidad de mantenimiento, ya que los sitios que utilicen tarjetas inteligentes, como teléfonos públicos y máquinas vendedoras, ya no serán objetivo de los ladrones, reduciendo considerablemente, la necesidad de realizar composuras y reemplazos.

Séptimo, se reduce el fraude y los tiempos de cada transacción, pues las operaciones se pueden realizar *off-line*, es decir, el emisor no tiene necesidad de conectarse a un servidor para consultar o pedir autorización de la transacción ya que la información necesaria está en la tarjeta. El equipo del emisor almacenaría cada transacción en la memoria y posteriormente, esa información sería recogida para actualizar el sistema central.

Octavo, es flexible, ya que poseen un entorno de programación que permite crear, almacenar o suprimir aplicaciones en ellas lo que significa que es posible hacer tarjetas inteligentes a la medida seleccionando para la tarjeta, las aplicaciones que se adapten a la circunstancias y necesidades de cada persona u organización. Además, es posible agregarle a la tarjeta una banda magnética, un código de barras, un panel de firma, la foto y la verificación por medios biométricos, lo que permite que las tarjetas sean personalizadas.

Por último, a través de Internet los usuarios de tarjetas inteligentes podrán comprar por computador y pagar por red con mayor seguridad, ya que la cantidad se debita de su saldo dentro de la tarjeta inteligente.

Una de las personas que más ha contribuido a incrementar la seguridad de las tarjetas inteligentes gracias a sus estudios, es el experto británico Ross J. Anderson; en su página de Internet personal, <http://www.cl.cam.ac.uk/users/rja14/>, se puede encontrar todos sus artículos sobre este tema.

Como principal inconveniente de las tarjetas inteligentes está la inversión adicional que tiene que hacer una organización para comprar y configurar la infraestructura de dispositivos lectores y las propias tarjetas; por otro lado, un usuario puede perder su tarjeta con relativa facilidad, y durante el tiempo que no disponga de ella no puede acceder al sistema.

También se deben establecer reglas especiales para evitar comprometer la seguridad (por ejemplo, marcar como tarjeta inválida en una base central de datos, para que un potencial atacante no pueda utilizarla). Además, tiene la desventaja de que depende de la energía eléctrica para su utilización, es vulnerable a los fluidos y es necesario contar con un lector para tarjetas inteligentes.

2.1.4. Tipos de tarjetas inteligentes

Se distinguen dos tipos principales:

- Tarjetas que sólo tienen memoria

Estas tarjetas permiten leer, escribir, borrar y agregar datos, y pueden verse como pequeños disquetes con posibles opciones de seguridad. Se utilizan para reemplazar a las monedas en los teléfonos, como medios de acceso prepagado u otras actividades similares.

Estas tarjetas no tienen una estructura de directorios ni de archivos, como las tarjetas de microprocesador, para almacenar la información, sino que ve la memoria como un simple arreglo de *bits* (o *bytes*). Lo único que se necesita saber para leer o escribir en ellas, es saber en qué posición (en qué *bit* o en qué *byte*) se encuentra el dato a leer o a almacenar.

- Tarjetas con microprocesadores

Son las que podemos considerar verdaderamente como tarjetas inteligentes, pues ellas tienen la capacidad de procesar la información que contienen. En ellas se puede leer, escribir, borrar y agregar datos, y manipular información dentro de la memoria de la tarjeta.

Estos añaden funciones de seguridad, permiten el control de aplicaciones múltiples y ofrecen la posibilidad de actuar como interfaz de comunicaciones. Similar a una computadora, cuenta con un sistema operativo que controla el puerto de entrada y salida, con características de seguridad incorporadas y cuenta con un disco duro que permite el almacenamiento de la información.

Lo que principalmente diferencia una tarjeta de memoria con una de microprocesador, es la forma y el grado de seguridad con la cual los datos son almacenados y manipulados. Dentro de la tarjeta con microprocesador, los datos son almacenados como en un disquete, se pueden definir directorios y archivos, en donde, para cada directorio o archivo se les define un encabezado o *header* que contiene el nombre y tipo de directorio o archivo, y se hace una referencia hacia las llaves o claves que se deben de presentar para leer o escribir. Las claves se encuentran en un archivo especial de claves. El encabezado puede ser programado de manera que un archivo o directorio no pueda ser leído nunca, como es el caso del archivo de claves que se utilizan en el proceso de autenticación entre la tarjeta y el lector, a la cual, ningún dispositivo que no sea el mismo microprocesador de la tarjeta, tiene capacidad de leer.

Otro aspecto importante, que diferencia las tarjetas de microprocesador con las de memoria, es que las de microprocesador tienen la capacidad de manejar varias aplicaciones dentro de una sola tarjeta, manejando para cada aplicación, claves propias para la aplicación o servicio, es decir, puede usarse una sola tarjeta para control de préstamos de textos, uso de parqueo, monedero electrónico para cualquier servicio, etc., y que para cada uno de estos servicios se tenga un grupo de claves distinto, es decir, dentro de cada directorio se puede crear un archivo de claves diferente de los demás.

Las tarjetas inteligentes con microprocesador cuentan con un sistema operativo (*Card Operating System: COS*) que maneja la memoria interna, implementa la seguridad y sirve de intérprete para los comandos que envía el usuario desde el programa de aplicación y la respuesta que le envía el lector. El COS, ocupa de 3 a 20 *Kbytes* de memoria ROM y se fija en la tarjeta, en el momento de su fabricación. La información personal que es de importancia para el portador de la tarjeta, se almacena en memoria EEPROM.

Por medio del sistema operativo (COS), el diseñador de la aplicación define la estructura de seguridad de la tarjeta, ya sea a nivel de directorios o a nivel de archivos, pudiendo darle una clave distinta a cada operación dentro del archivo, con distinto tipo de acceso (lectura, escritura, etc.) y fijándola en el proceso de personalización o inicialización.

Usualmente, el programa que quema (personaliza o inicializa) el mapa es el que envía a la tarjeta las claves que se van a usar y se pueden generar de tres formas. Una opción es que el valor de las claves sean indicadas explícitamente, y son llamados *hard-coded*. Otra opción, común, es que las claves sean leídas de otra tarjeta maestra, que tiene allí grabadas las claves, pasando por un aparato llamado Octagon, que impide que las claves puedan ser conocidas por terceros.

La tercera y la opción más usual, es que haya una combinación, y es lo que se conoce como claves diversificadas, en donde la clave que cada tarjeta tiene, se construye siguiendo un algoritmo (arbitrario) a partir de una clave predefinida (una clave madre única para cada sistema) y de un dato disponible en dicha tarjeta (como su número de serie). Por ejemplo, si existen 3 empresas con sistemas basados en tarjetas, cada sistema posee una clave madre, de la cual, se generan todas las claves, una para cada uno de los usuarios de su propia empresa.

De esta manera, cada tarjeta tendrá una clave diferente, aunque construida a partir de la misma, de modo que si alguien logra abrir una tarjeta, solamente obtiene las claves de un usuario y no las de todos los usuarios del sistema, y normalmente los medios utilizados para descubrir las claves de una tarjeta, destruyen la misma por lo que, la información averiguada no servirá para realizar operación alguna con esa tarjeta.

Una vez se defina la estructura de directorios y archivos (el mapa de la tarjeta) dentro de la tarjeta con microprocesador, dependerá del COS y los permisos que den las claves de seguridad, que ésta se pueda modificar, si no lo permiten, no se podrá modificar o solamente se podrán agregar más archivos, por lo que es necesario poner especial cuidado al determinar la estructura de los datos en este tipo de tarjetas (con microprocesador).

Se tienen 3 parámetros principales para controlar el acceso a la información contenida en las tarjetas:

- Saber quién puede acceder a la información (todos, sólo el dueño de la tarjeta o solamente la persona/entidad que extiende la tarjeta).
- Saber cómo se puede acceder la información (sólo lectura, sólo para agregar información, para actualizar o no tiene acceso).
- Saber a qué dato se puede acceder de una manera específica. Para cada archivo, se puede definir para cada operación (lectura, escritura, etc.) la protección a usar (cuál llave, siempre, nunca).

En los casos en el que las claves de acceso no son suficientes para asegurar la integridad de la información, esta puede cifrarse con un método especial. Para obtener un mayor grado de seguridad, debe existir un proceso de autenticación que asegure que se estén usando las tarjetas y las computadoras autorizadas para acceder la información.

El proceso de autenticación, sirve para determinar si el usuario que carga la tarjeta es el dueño de la misma y se hace uso de claves que permiten tanto a la tarjeta como a la aplicación determinar ésto. También se puede almacenar dentro de la tarjeta, la huella digital, la imagen de la retina o del iris de la persona, la modulación de la voz del dueño de la tarjeta o el patrón de su firma, para asegurar que la persona que usa la tarjeta, es la dueña de la misma.

Existe un mecanismo llamado Mensaje Seguro que encripta la comunicación entre la tarjeta y el dispositivo de lectura, de modo que la información que se transmite, no pase en claro por los contactos. Nunca se transmiten las claves, sino que se transmiten mensajes cifrados que permiten lo que se conoce como Autenticación Mutua, es decir, la tarjeta y el dispositivo lector se demuestran mutuamente, que conocen las claves.

El proceso de autenticación permite determinar si una terminal o PC está comunicándose con una tarjeta válida para una aplicación en específico, por ejemplo, una tarjeta de teléfono público usado en México no es válida para los teléfonos públicos de Guatemala. Este es un proceso de dos fases, y para ello hay dos comandos: *InternalAuthentication* y *ExternalAuthentication*. Ambos casos se basan en que ambas partes deben conocer algo en común, una clave especial, diferente de las claves de cada aplicación (o directorio). Esto sirve para que esta clave, conocida como *calculation key*, no se bloquee cuando se intente escribir sobre una tarjeta en la cual no se tiene permiso para ello. La tarjeta envía al lector un número pseudo-aleatorio, conocido como *Challenge*. La terminal o PC que lee, calcula un criptograma con ese número y el valor de la *calculation key*, y lo envía a la tarjeta. La tarjeta, hace el mismo cálculo, y compara su criptograma con el que recibió de la terminal. Si coinciden, entonces sabe que es una terminal válida, y la terminal sabe que encontró una tarjeta válida.

Los elementos de seguridad físicos del *chip* hacen que la tarjeta se destruya si alguien intenta forzar la lectura del contenido de la tarjeta, o lo somete a rangos de voltaje o frecuencias de reloj distintos a los que maneja, de modo que, la tarjeta que esté bien diseñada deberá de fallar automáticamente en respuesta a tales condiciones. Además, borra la memoria por radiación ultravioleta si se intenta abrir un módulo para explorarlo, utiliza métodos como los *bits* centinelas que permiten detectar un uso ilegal, el consumo aleatorio de corriente independiente de la operación que se esté realizando y del número de *bits* escritos y hace uso de memorias resistentes a campos magnéticos y electromagnéticos.

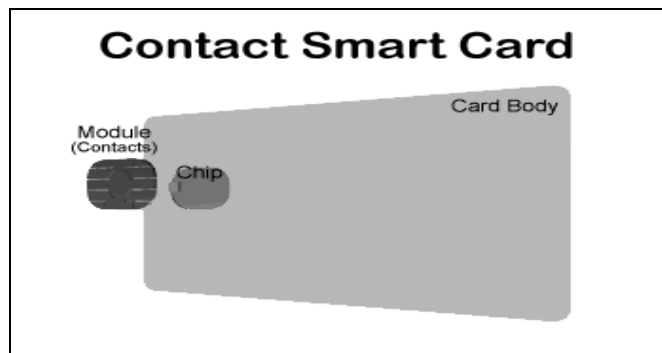
En las tarjetas de microprocesador, luego de una cantidad determinada de inserciones no válidas en el dispositivo de lectura (cantidad que se le indica al definir el mapa de la tarjeta), la clave correspondiente a la aplicación que trata de leerlo queda invalidada, por lo que cualquier dato que esta protegía queda inaccesible, pero los datos protegidos por otras claves no se ven afectados. En las tarjetas de memoria, algunas se desactivan luego de tres pruebas inválidas, otras tienen una llave de desbloqueo y otras, las más comunes, no tienen ninguna protección.

En algunos casos, los circuitos son diseñados de manera que se tornen inoperables una vez que la tarjeta haya sido programada, para que esa información personal no pueda ser alterada. Los fabricantes también emplean técnicas especiales de protección que impiden a los ladrones ver directamente el circuito microscópico de la tarjeta. Además, el *chip* se adhiere al sustrato con un epóxico que rompe el procesador si se intenta extraer el *chip*, tanto en las tarjetas de memoria como en las de microprocesador.

También se pueden clasificar las tarjetas inteligentes, según el medio que se utilizará para acceder físicamente, a los datos almacenados en ellas:

- Tarjetas inteligentes de contacto, que deben ser insertadas en un lector de tarjetas inteligentes para que haga contacto con conectores eléctricos encargados de transferir datos desde y hacia el *chip*.

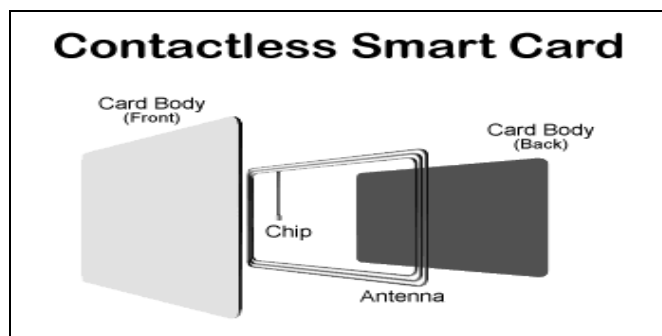
Figura 3. Tarjeta inteligente de contacto



Fuente: www.gemplus.com/basics/what.htm

- Tarjetas inteligentes sin contacto (inalámbricas), deben pasarse cerca de una antena para poder llevar a cabo una transacción. Tienen un microchip electrónico y una antena incrustada en su interior, que les permite comunicarse con otra antena o aparato de conexión sin necesidad de un contacto físico. Existen tarjetas que operan a una radio frecuencia de 13.56 MHz y a 125kHz, estos son ideales para usarlos en transacciones que deben ser procesados de manera rápida, como el tránsito masivo de personas en un lugar.

Figura 4. Tarjeta inteligente sin contacto



Fuente: www.gemplus.com/basics/what.htm

- Tarjetas mixtas o híbridas, combinación de las tarjetas de contacto junto con las que son sin contacto (combinación de ambos protocolos de comunicación).

2.1.5. Estándar ISO 7816

Uno de los motivos principales por el que no se han adoptado rápidamente las tarjetas inteligentes fuera de Europa, es la incompatibilidad de las aplicaciones, tarjetas y lectores. Por ello, es necesario que exista interoperabilidad entre los productos de los diferentes proveedores, dando lugar a una mayor aceptación de los consumidores de tarjetas inteligentes y para que las organizaciones distribuyan tarjetas inteligentes para su uso dentro de la empresa.

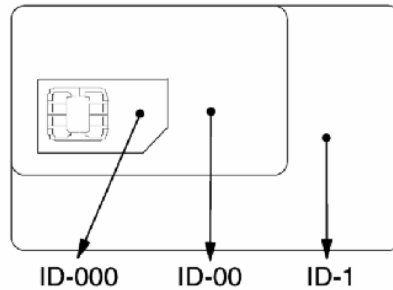
Para promocionar la interoperabilidad entre las tarjetas inteligentes y los lectores, la Organización Internacional de Estándares (ISO) desarrolló los estándares ISO 7816 para las tarjetas de circuitos integrados con contactos. Estas especificaciones se centraron en la interoperabilidad física, eléctrica y de protocolo de vinculación de datos, así como también, rangos de temperatura y flexibilidad, y cómo debe darse la comunicación entre el microchip y el mundo exterior.

Entre los estándares definidos podemos mencionar algunos como, el estándar ISO 7816 – 1 que define las características físicas de la tarjeta, el ISO 7816 – 2 que define la dimensión y la posición de los contactos (*chips*) en la tarjeta, el ISO 7816 – 3 que define la señal electrónica escogida y los protocolos de transmisión, y el ISO 7816 – 4 que define aspectos como la estructura de los paquetes llamados APDU's.

El estándar ISO 7816 define, entre muchas otras, las siguientes características físicas:

- Se definen 3 tamaños para las tarjetas: ID-1 (el más habitual), ID-00 e ID-000 (el utilizado para GSM)

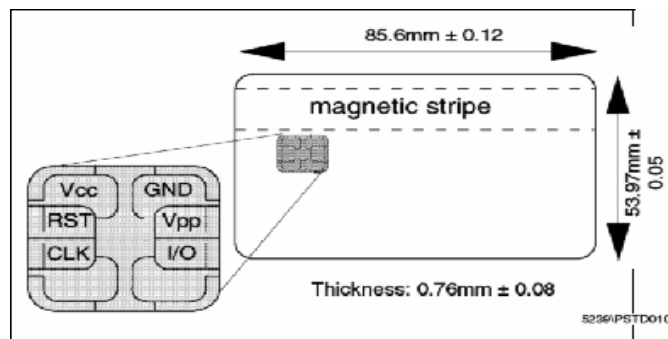
Figura 5. Tamaños de tarjetas inteligentes



Fuente: www.dte.us.es/tec_inf/itis/peri_int/trabajos/TarjetasInteligentes.ppt

- La tarjeta debe resistir ataques con rayos X y luz ultravioleta, tener una superficie plana, permitir cierto grado de torsión, resistir altos voltajes, campos electromagnéticos y electricidad estática, y no disipar más de 2.5W.
- Se define la dimensión y localización de los contactos. Existen ocho contactos, de los cuales, sólo se usan seis: Vcc, Vpp, GND, RST, CLK e I/O

Figura 6. Dimensión y localización de los contactos



Fuente: www.dte.us.es/tec_inf/itis/peri_int/trabajos/TarjetasInteligentes.ppt

- Se define las señales electrónicas y protocolos de comunicación: protocolo T=0 ó protocolo orientado a carácter, el más usado (GSM), requiere dos transacciones para recibir datos; protocolo T=1 ó protocolo orientado a bloque, requiere una única transacción para recibir datos.
- La tarjeta deberá trabajar correctamente y no tener ninguna grieta o cuarteada después de 1000 flexiones.

En 1996, Europay, MasterCard y VISA (EMV) definieron una especificación de tarjetas inteligentes propia para el sector de la industria de servicios financieros, adoptando los estándares ISO 7816 y definiendo algunos tipos de datos y reglas de codificación adicionales específicas para su uso en esta industria. Para el Sistema Global para las comunicaciones móviles (GSM), también se adoptaron los estándares ISO 7816 en su especificación de tarjetas inteligentes, para habilitar la identificación y autenticación de usuarios de teléfonos móviles.

En mayo de 1996, se formó el grupo de trabajo PC/SC (*Personal Computer/Smart Card*) en asociación con las grandes compañías de equipos informáticos y tarjetas inteligentes: Bull, Hewlett-Packard, Microsoft, Schlumberger y Siemens Nixdorf. El fin de este grupo es el desarrollo de especificaciones para solucionar los problemas de interoperabilidad, publicando la primera versión en diciembre de 1997, en la dirección <http://www.smartcardsys.com/>. Con el pasar del tiempo, se han unido Gemplus, IBM, Sun Microsystems, Toshiba y Verifone a este grupo de trabajo PC/SC.

2.1.6. Lectores de tarjetas inteligentes

Dispositivos que actúan como *interface* entre el usuario y el sistema, es decir, conectan una tarjeta inteligente al equipo del usuario. Pueden ser alámbricos, inalámbricos, con teclado, sin teclado, con o sin pantalla. Sus capacidades varían de acuerdo a las necesidades de los usuarios.

Existen lectores similares a los de los discos magnéticos, que se instalan dentro del CPU, otros son terminales portátiles diseñadas para que sean sujetadas con la mano, algunos son capaces de tomar la huella digital del usuario, es decir, traen un escáner de huellas digitales.

También se pueden encontrar teclados con lectores de tarjetas incorporados, y teclados con lectores de tarjeta y escáner de huellas digitales incorporado. Por otro lado existen lectores diseñados para integrarse a cualquier dispositivo electrónico, por ejemplo, en un aparato creado para habilitar el ingreso de vehículos a un parqueo.

Figura 7. Lector tipo *driver*



Fuente: www.gemplus.com/products/hardware/gempc410_fd.htm

Figura 8. Lector con conexión USB



Fuente: www.gemplus.com/products/hardware/gempc430.htm

Figura 9. Lector de tarjetas inteligentes sin contacto



Fuente: www.gemplus.fr/developers/products/gemeasyaccess608/index.htm

Figura 10. Lector de tarjetas inteligentes para integrar a cualquier dispositivo electrónico



Fuente: <http://www.gemplus.com/products/gemcore410/>

Figura 11. Lectores con escáner de huella digital ACS



Fuente: www.smartcardfocus.com

Figura 12. Teclado con lector de tarjeta inteligente y escáner de huellas digitales incorporado



Fuente: http://CardLogix.com/product_readers_pbiometrics.asp

Primero se debe elegir el tipo o tipos de tarjetas a usar, y luego se escoge el aparato lector. Básicamente existen dos tipos:

- **Universales:** permiten leer más de un tipo de tarjeta, suelen ser caras y son útiles en entornos en que los diferentes usos de las tarjetas implique la utilización de diferentes tipos de estas (acceso a zonas, expedición de moneda, etc.).

- **Especializadas:** pueden leer unos pocos tipos de tarjetas similares. Debido a que su funcionamiento es más simple, es más fácil conseguirlas. Generalmente, van conectadas a un computador.

La mayoría de los lectores se conectan al equipo a través de un puerto serie RS-232, una ranura PCMCIA (*Personal Computer Memory Card International Association*) de tipo II o un puerto USB (bus serie universal).

Los lectores presentan el inconveniente de que la comunicación entre tarjeta y lectora es propietaria, es decir, el tipo de comunicación que maneja es específicamente para ese modelo de tarjeta y ese modelo de lector dificultando así, la forma de comunicar cualquier programa con los distintos modelos de lectores existentes, para solucionar esto, están las especificaciones OCF (en entornos Java), PC/SC (en entornos Windows) y MUSCLE (en entornos Linux), que son API's que nos ayudan con el proceso de comunicación entre la aplicación o programa y el lector, de modo que el programador sepa qué comandos usar, cómo pasarle los parámetros (cómo empaquetarlos en un APDU) y cómo interpretar los APDU's (paquetes de información) que le envíe el lector, todo lo cual, depende del COS (sistema operativo de la tarjeta) y/o el modelo de la tarjeta.

La estructura del APDU está definida en el estándar ISO 7816-4, pero los valores de algunos de sus parámetros, dependen del COS del proveedor o fabricante del modelo de tarjeta en específico, información que algunos publican, otros exigen la firma de un documento y otros venden *kits* de desarrollo que suelen ser caros, pero sin esa información, es muy difícil trabajar con las tarjetas.

2.2. Botones (*i-buttons*)

Otro método para auto-identificarse, es por medio de dispositivos de memoria de contacto, conocidos como los *iButton*. Con más de 20 millones de botones en circulación, la tecnología de los *iButton* se ha situado en todos los continentes. El *iButton* fue inventado y es fabricado exclusivamente por la empresa *Dallas Semiconductor*, utilizado principalmente en ambientes que demandan muchos cuidados, y tiene funciones de identificación y autenticación, tomar la hora y el día de eventos, cuidar pertenencias e inventariar.

2.2.1. Definición

Un *iButton* es un *chip* de computadora de 16mm. dentro de un empaque de acero inoxidable con un tamaño de 17.35mm x 3.1mm a 5.89mm (dependiendo de la función que tenga), dentro del cual, las personas pueden transportar información actualizada, con la que se pueden realizar diversas transacciones. La energía que requiere para operar, lo puede obtener ya sea desde la línea de datos (energía parasitaria) o desde una batería de litio que es parte del botón.

Figura 13. *iButton* de 4kbit de EEPROM



Fuente: store.iButton.com

El botón es de acero resistente y durable para que una persona lo lleve todos los días en sus accesorios, tales como un anillo, un llavero, la billetera, el reloj o una placa de identificación.

Figura 14. Anillo para botones JAVA



Fuente: store.iButton.com

Figura 15. Llavero para botones



Fuente: www.iButton.com/iButtons/index.html

Los botones se han vendido, en cantidades por debajo de las tarjetas inteligentes, debido a que ya existen muchos usuarios que tienen como base las tarjetas inteligentes, son relativamente un poco más caros que las tarjetas (costo de botones desde \$.1.32 a \$.6.89 dependiendo del modelo y la capacidad del botón, en este año), ya que *Dallas Semiconductor*, la empresa fabricante de los *iButtons*, no ha permitido que otras empresas los fabriquen.

Los *iButtons* requieren de una conexión con un aparato lector, como el *Blue Dot Receptor* (un producto con un costo de \$15 para este año), para realizar las operaciones de lectura y escritura en el botón. A diferencia de las tarjetas inteligentes, no existen los botones sin contacto, es decir, siempre requieren de un contacto físico con un aparato lector.

Figura 16. Lector *Blue Dot* con conexión al puerto paralelo para botones



Fuente: www.iButton.com/iButtons/blue_dot.html

Entre sus ventajas, encontramos que la información puede agregarse fácilmente, la transferencia de la información es libre de las restricciones de las redes, y además, es reusable. Tiene ventajas sobre las tarjetas inteligentes respecto a la durabilidad y longevidad que tiene (aproximadamente de 10 años) y es fabricado con un material durable, el acero inoxidable, lo que lo hace resistente a caídas, ralladuras, apretones y al agua, y le da la habilidad de sobrevivir a temperaturas entre los $-40\text{ }^{\circ}\text{C}$ y los $+70\text{ }^{\circ}\text{C}$.

A los botones se les puede dar un formato con un directorio similar a los disquetes, que le permite acceder directamente a la información que se desea y evitando que se modifiquen otros datos almacenados dentro del botón.

Con respecto a la seguridad, se puede tener 4 niveles de seguridad con los *iButtons*:

- Nivel 1: todos los *iButton* traen su identificador serial único de 64 *bits* que puede ser usado como clave de seguridad o ID, y como clave única para encriptar datos.
- Nivel 2: el modelo DS1991L *MultiKey iButton*, tiene 3 bloques de 48 *bytes* de datos que soporta hasta 3 aplicaciones distintas, en donde cada bloque de datos puede estar protegida por una clave de 8 *bytes*. El modelo DS1977 EEPROM *iButton* de 32 *kbytes*, está organizado en páginas de 64 *bytes* cada una con opción a protección por medio de claves de 64 *bits*.

- Nivel 3: los botones SHA (*Secure Hash Algorithm*) *iButtons*, modelos DS1961S y DS1963S, están diseñados especialmente para transacciones de monedero electrónico pero pueden ser usados para autenticación en cualquier aplicación, y se basa en algoritmos hash.
- Nivel 4: los botones *Java-Powered Cryptographic iButton*, son los dispositivos más seguros, ya que, antes de revelar su contenido, lo borra en respuesta a cualquier intento de forzarlo a revelarlo.

2.2.2. Aplicaciones

Los botones son ideales para cualquier aplicación que requiera que la información viaje con una persona u objeto. Permite al portador de la misma, acceder a un edificio, un computador, una pieza de un equipo, o a un vehículo. También puede servir para la medición de procesos en busca del aumento de la eficacia en los mismos, como la producción, entregas y mantenimientos, o bien, pueden ser utilizados como dinero para transacciones pequeñas.

Existen modelos de botones que cuentan con un reloj de tiempo real y con la capacidad de medir temperaturas, se puede almacenar más de 8 *kbytes* de información, permite grabar más de 2,048 lecturas de temperatura en intervalos ajustables de 1 a 255 minutos, por lo que estos son recomendados para aplicaciones médicas o cualquier aplicación sensible a la temperatura. Los botones pueden almacenar datos personales, médicos y otros dependiendo de las necesidades de las personas, lo que lo hace un dispositivo con abundantes aplicaciones.

Los botones han sido utilizados como monedero electrónico, en Turquía donde han tenido su mayor éxito, se usa para el sistema de tránsito como tarjeta de débito para andar en buses, trenes y transbordadores; en Argentina y en Brasil los usan en parquímetros; en México y Moscú los usan las gasolineras; y en los Estados Unidos los utilizan para aumentar la eficacia del servicio postal al pegar un botón en la esquina de cada buzón de correo para optimizar su trayectoria.

Otros usos que se les han dado a los botones son:

- Mantenimiento: la empresa *Ryder*, almacena la información sobre el mantenimiento de sus vehículos dentro de botones pegados al exterior de sus camiones.
- Tiempo y asistencia: agricultores han usado botones para rastrear el monto del trabajo completado.
- Control de acceso: uso de botones como llaves electrónicas para el acceso a construcciones, habitaciones, bodegas, áreas restringidas, etc. En Nueva York, están siendo utilizadas en más de 10,000 edificios con la marca comercial *ComKey*.
- Información médica: la información médica personal de las personas, es almacenada en brazaletes de botones en Australia.
- Comercio electrónico: los botones han sido aprobados como el medio para contener estampillas electrónicas para el servicio postal de los Estados Unidos. Las estampillas pueden ser adquiridas vía Internet.
- Control de procesos: la compañía Ford almacena los datos de calibración de los motores en botones adheridos a cada motor.
- Monitoreo de la temperatura: en Australia ha sido utilizado por empresas de servicio de banquetes para monitorear la temperatura de la comida. En Estados Unidos, se ha utilizado en un proyecto piloto para guardar el historial de fluctuaciones de temperatura en los nidos de pájaros cantores durante su incubación, con el fin de hacer investigaciones ecológicas para la conservación y mantenimiento de los mismos.

- Monedero electrónico: en Denver, Estados Unidos, se han colocado máquinas de venta de bocadillos y bebidas, en los que no se utilizan monedas sino botones para adquirir los productos.

En caso de que las personas no quieran desarrollar sus propias aplicaciones, existen las compañías llamadas *Authorized Software Developers (ASDs)*, que desarrollan soluciones completas para sistemas basados en botones. Además, estas compañías pueden desarrollar aplicaciones a la medida de los clientes.

2.2.3. Tipos de botones

Los botones tienen garantizado un registro numérico único grabado en el silicio (del *chip* que está dentro del empaque) que sirve como identificador y clave de acceso de los mismos. Los *iButton* se pueden clasificar en 5 tipos:

- **Address only.** El modelo DS1990A, que es el modelo base, contiene 64 *bits* de memoria de sólo lectura (ROM) y es usado para aplicaciones como llave de control de acceso, identificador de equipos para su inspección y mantenimiento, identificador de puntos de ubicación o lugares para revisión de rutas a tomar.
- **Memoria.** Usados para lectura y escritura, capaz de almacenar textos y fotos digitales. Los botones NV RAM se pueden encontrar con capacidades de 1 *kbits* a 64 *kbits* y son utilizados para aplicaciones en los que se necesita actualizar los datos con una gran frecuencia. Los botones EPROM van desde 1 *kbits* a 64 *kbits* y se usan para aplicaciones donde los datos deben ser permanentes, es decir, no se deben borrar. Los botones EEPROM van desde 256 *bits* a 32 *kbytes* y se usan en aplicaciones donde se necesita una frecuencia menor de actualización de datos que las aplicaciones que necesitan las NV RAM, son más baratas que las NV RAM pero tiene un límite menor de escrituras.

- **Reloj de tiempo real.** Utilizado para registrar la hora actual, por ejemplo, para tomar la hora de entrada y salida de los trabajadores de una empresa y luego en base a los datos guardados, realizar los pagos. También son usados para rastrear el número de horas en el que un sistema es encendido para propósitos de mantenimiento y autorización.
- **Seguridad.** Son una alternativa a las tarjetas inteligentes y pueden usarse para control de acceso físico, monedero electrónico, autorización de *software*, autenticación de dispositivos, control de acceso de usuarios a una PC o a un servidor. Los botones operados por Java, cuentan con un microprocesador y un acelerador aritmético de alta velocidad, que genera los números necesarios para encriptar y desencriptar información, permite a los programadores en Java a introducirse a poderosas herramientas de desarrollo para incrementar la cantidad de aplicaciones y que corran más rápidamente.
- **Temperatura.** Diseñados para tomar la temperatura de productos de los cuales se desee garantizar su calidad y frescura. Integran un termómetro, un reloj calendario, un historial de temperaturas y 512 *bytes* de memoria adicional para almacenar información, de envíos por ejemplo. Toma la temperatura del ambiente en el que ha sido instalado, tomando desde los -40 C° a los $+125\text{ C}^{\circ}$ dependiendo del modelo escogido. Es resistente al agua pero no impermeable. Duran 10 años o 1×10^6 (1,000,000) mediciones de $50\text{ }^{\circ}\text{C}$ o menos (lo que se dé primero).

3. ANÁLISIS DE LOS SISTEMAS

Antes de proponer un nuevo sistema para el control de préstamos de textos para las bibliotecas, se analiza el estado actual del sistema de préstamos y se compara con un análisis del sistema propuesto, con el fin de determinar cuáles serían los resultados de la implementación del nuevo sistema.

3.1. Ventajas y desventajas del sistema actual

3.1.1. Ventajas

- Todos los encargados de las bibliotecas conocen perfectamente el sistema de préstamos que se está usando.
- No dependen de las computadoras para dar su servicio.
- Solamente se invierte en hojas de papel, calcomanías y lapiceros.

3.1.2. Desventajas

- Otras entidades dentro de la universidad, distintas a la de las bibliotecas, no pueden saber si una persona debe textos en alguna biblioteca.
- El carné es retenido por la biblioteca en el momento del préstamo, lo que limita al estudiante a no realizar cualquier otro préstamo o trámite en el que necesite su carné.
- Hasta el momento, los bibliotecarios no podrán saber si los estudiantes pertenecen a su facultad o si estarán o no inscritos el próximo año (2005), ya que el carné actual, tiene vigencia solamente para este año (2004).

- Los encargados de las bibliotecas no saben quién realizó un préstamo a alguien que no haya devuelto los textos, provocando así, que ellos tengan que cargar con el costo de los mismos y pagarlos en caso de extravío o robo.
- Para que un estudiante pueda inscribirse, es necesario verificar la solvencia del estudiante, obligando a que éste tenga que ir tanto a la Biblioteca Central como a la biblioteca de su facultad para que le otorguen una constancia de que estén solventes, constancia que puede no concordar con la realidad si horas después de solicitarla, presta textos y luego se va a realizar el trámite de inscripción.
- Para evitar que los estudiantes se lleven sin permiso los textos a sus hogares, tienen políticas estrictas, que pueden llegar a ser molestos para los estudiantes.

3.2. Descripción general del sistema inteligente propuesto

El sistema inteligente que se propone en este trabajo, vendría a complementar el área de control de préstamos, y se ocuparía exclusivamente de ver quién tiene prestado textos de cada biblioteca, ver si un usuario tiene moras en otras bibliotecas y proporcionar los datos del usuario de manera automática al sistema, y en el momento que se desee, se podrá conocer la solvencia de un usuario. De esta manera, cualquier entidad autorizada, podrá conocer la solvencia en las bibliotecas de la Facultad de Ingeniería sin necesidad de que el usuario tenga que ir a cada biblioteca para solicitar su solvencia.

Básicamente, se almacenarán dentro del carné inteligente los datos generales del usuario (nombre, dirección, teléfono, carrera, año de última inscripción) y los datos de los textos que tenga prestados en cualquiera de las 4 bibliotecas que forman parte del nuevo sistema, de modo que cualquier biblioteca podrá saber cuántos y qué textos tiene prestados el usuario en un momento determinado, y saber si tiene alguna mora.

Al tener los datos generales del estudiante dentro del carné, el sistema inteligente podrá obtener los datos del usuario a través de él y alimentar de manera automática su base de datos, sin tener la necesidad de que el usuario llene formularios y luego se tenga que ingresar al sistema manualmente.

Si bien, cada biblioteca podrá ver qué textos de otras bibliotecas tiene prestados el usuario, solamente la biblioteca que es dueña del texto prestado, podrá realizar alguna modificación respecto al texto referido, es decir, una biblioteca “X” no podrá grabar en el carné que fue devuelto el texto de una biblioteca “Y”, puesto que el texto no pertenece a su biblioteca.

3.3. Ventajas y desventajas del sistema inteligente propuesto

3.3.1. Ventajas

- Cualquier bibliotecario obtendrá los datos generales del usuario que preste un texto, como su nombre, dirección y teléfono, para cualquier reclamo, sin necesidad que el usuario llene formularios, ya que el carné inteligente le proporcionará estos datos de manera automática.
- Cualquier otra entidad, distinta a las bibliotecas, que cuente con un aparato lector y que esté autorizada para hacerlo, podrá conocer en cualquier momento, si el estudiante debe algún texto en alguna biblioteca, para cualquier propósito (Ej.: para saber si tiene derecho a inscribirse o no, o para determinar si se le prestará un texto o no).
- Se automatizaría el proceso de verificación de datos de los estudiantes cada vez que requieran el préstamo de textos, pues esta verificación, la realizaría el sistema inteligente, minimizando así, el tiempo de atención para cada estudiante.

- Los bibliotecarios podrán saber en cualquier momento, qué usuarios están morosos, puesto que contarán con una base de datos sobre los préstamos que hayan realizado.
- Se podrá mantener un control sobre quién es la persona responsable de verificar los datos de los estudiantes antes de prestarles los textos, evitando que personas que no tuvieron que ver con préstamos no devueltos, tengan que pagar el costo de los textos no devueltos.
- Permitirá a los estudiantes, contar en todo momento con su carné universitario para cualquier trámite que desee realizar con él.
- Se puede llevar un mejor control sobre la cantidad de textos que presta cada estudiante.
- No es necesario que todas las bibliotecas estén en red para conocer la solvencia de los usuarios, permitiéndoles decidir si se les presta o no, un texto.
- El carné inteligente podría utilizarse no sólo para el control de préstamo de textos en bibliotecas sino también en cualquier otra aplicación como monedero electrónico para pago de fotocopias, pago de retrasadas o cursos de vacaciones, o como tarjeta de control de acceso al parqueo, entre muchas otras posibilidades.

3.3.2. Desventajas

- La introducción de una nueva forma de procesamiento en cualquier área de un sistema, puede dar lugar a la resistencia al cambio por parte de los que deben interactuar con él.

- Se necesita invertir en equipo para cada una de las bibliotecas, y mantener así, la integridad de la información sobre los préstamos que hacen los estudiantes, y la eficiencia del sistema.

3.4. Conclusión

Las bibliotecas se han visto obligadas a adoptar políticas restrictivas para el préstamo de textos, aún cuando los estudiantes solamente necesitan sacar los textos de la biblioteca más no de la universidad. Muchas veces, tratar de mantener ese control, no les ha dado los resultados esperados y las bibliotecas, al presentarse casos en los que los usuarios no devuelven los textos en el momento debido, terminan con una cantidad deficiente de copias por texto. Esto, no sólo resulta en un servicio que no satisface las necesidades de los usuarios al no encontrar los textos que necesitan, sino también en el desgaste que provoca a los bibliotecarios tratar de mantener ese control y evitar estas situaciones.

Si bien existen desventajas en relación al nuevo sistema propuesto, acerca de los costos para implementarlo, éste ofrece muchas ventajas que a largo plazo disminuirán los costos de las bibliotecas, pues no tendrían que costear los textos no devueltos a causa del desconocimiento de los antecedentes de las personas que llegan a solicitar el servicio de préstamo. Además, permitiría a los bibliotecarios, conocer exactamente qué textos de la biblioteca tiene cada usuario, cuántos y cuándo tuvo que haberlos devuelto, obtener los datos del usuario y así poder tomar las medidas correspondientes.

Por otro lado, ofrece la posibilidad de automatizar su sistema de préstamos, minimizando el tiempo de atención para cada estudiante y el tiempo de verificación de antecedentes de los estudiantes para realizar los préstamos.

La desventaja del nuevo sistema, acerca de la resistencia al cambio que puedan experimentar los bibliotecarios, puede ser superada si se informa acerca de todas las ventajas que puede traerle, como evitarle grandes papeleos para poder verificar la solvencia de los estudiantes, y evitarle el riesgo de tener que pagar el costo de libros no devueltos.

Se aclara, que este sistema no trata de cambiar las políticas de préstamo que existe en cada biblioteca, sino que trata de atenuar el problema que se tiene actualmente para controlar los préstamos de textos a estudiantes y catedráticos al otorgarles un carné único para préstamo de textos en las bibliotecas con la cual, las bibliotecas tendrán la capacidad de obtener, de manera automática, los datos de los usuarios sin que ellos tengan que llenar formularios largos con sus datos y evitando que se presten más textos de los permitidos en cada biblioteca, a cada persona que solicite el servicio.

4. SELECCIÓN DEL DISPOSITIVO ADECUADO

Para calcular la cantidad de memoria que se necesita, hay que determinar los requisitos de espacio de:

- El espacio necesario para almacenar los datos requeridos para la aplicación.
- El espacio para las aplicaciones que se puedan incorporar en el futuro.

4.1. Cuantificación del tamaño necesario para cada dato

Los datos que desean saber los bibliotecarios acerca de los estudiantes, son:

- Nombre
- Carné (ya sea de estudiante, profesor o personal administrativo)
- Dirección
- Teléfono
- Estar inscrito en el año en curso, en la Universidad de San Carlos de Guatemala
- Facultad (carrera) a la que pertenece
- Que no tenga más de 2 ó 3 libros (dependiendo de cada biblioteca) prestados

Para cada texto:

- Saber la fecha en que deben ser devueltos
- Saber qué texto están prestando.
- Saber quién se lo prestó

Ya que las bibliotecas de ingeniería cuentan con el programa Winisis para tener un inventario de los textos con los que cuenta y para que los usuarios realicen la búsqueda de los textos que deseen, se propone usar un número correlativo para identificar cada texto, tomando el número MFN que crea Winisis, debido a que el número de clasificación de los textos utilizado, difiere en ciertos aspectos en cada biblioteca, además, el número de clasificación de los libros, sigue el Sistema de Clasificación Decimal de Dewey, el cual puede llegar a tener una longitud demasiado grande, haciendo el uso del espacio dentro del dispositivo inteligente, poco eficaz.

Cuando almacenamos datos en un dispositivo, debemos de procurar aprovechar al máximo el espacio que tenemos para almacenarlos, y para ello, es necesario que conozcamos el espacio que ocupan los datos dependiendo del tipo de dato que este sea. A continuación, se presenta un cuadro con los tipos de datos básicos del lenguaje de programación C++, indicando el espacio en *bytes* que ocupan, dependiendo del tipo de dato que sea.

Tabla I. Espacio en *bytes* utilizado por cada tipo de dato.

TIPO DE DATO	DESCRIPCIÓN	BYTES
<i>Char</i> (carácter)	Caracteres ASCII, y números enteros comprendidos entre -128 y 127. Por ejemplo: a, 7, +.	1
<i>Int</i> (enteros)	Números enteros que estén comprendidos entre -32768 y 32767	2
<i>Unsigned long int</i>	Números enteros comprendidos entre 0 y 4,294,967,295	4
Float	Números con decimales comprendidos entre $3.4 \cdot 10^{-38}$ y $3.4 \cdot 10^{38}$. Por ejemplo: 22277.326	4

Fuente: Schildt, Herbert. **C++ para programadores**. Pág. 59.

Para aprovechar el espacio de la memoria del dispositivo al máximo, los datos pueden ser almacenados en formato binario, es decir, a nivel de *bits* en donde cada *bit* puede almacenar un valor cero (0) o uno (1).

Este formato permite que, dependiendo de la cantidad de *bytes* que se asignen para representar un dato (un campo) de la información que necesitemos almacenar, podamos representar $2^{(n*8)}$ posibles combinaciones de números que representen el dato a almacenar, donde n es la cantidad de *bytes* asignados para ese campo, y se multiplica por 8 ya que un *byte* consta de 8 *bits* y cada *bit* puede almacenar un 0 ó un 1 en formato binario. Por ejemplo, si asignamos 1 *byte* para almacenar a qué facultad pertenece un estudiante, podríamos identificar hasta $2^{(1*8)} = 256$ facultades distintas, con la combinación correcta de ceros y unos en el *byte* asignado para ese propósito.

A continuación, se presenta un cuadro en el que se expone la cantidad máxima de datos distintos que se puede representar, dependiendo de la cantidad de *bytes* asignadas.

Tabla II. Valores máximos que se pueden representar con cierta cantidad de *bytes*

<i>BYTES</i>	VALOR MÁXIMO QUE SE PUEDE REPRESENTAR
1	256
2	65,536
3	16,777,216
4	4,294,967,300
5	1,033,511,600,000

A continuación, se presenta una propuesta del espacio estimado para los datos a almacenar, tanto el mínimo necesario para usar el formato binario, como un máximo si se almacena un dígito como caracter tipo char (al que se le llamará a continuación, formato caracter), en el caso de que se tenga disponible toda la capacidad del dispositivo para implementar el sistema, ya sea porque el carné no se vaya a utilizar para ninguna otra aplicación o servicio, o porque se utilicen tarjetas o botones de mayor capacidad.

- Datos generales de la tarjeta y el usuario dueño del carné:

Tabla III. Espacio estimado para almacenar los datos generales del usuario

Dato	Formato binario (<i>bytes</i>)	Formato caracter (<i>bytes</i>)
ID del carné – ID emisor (dividido en año y correlativo, en donde para el año sólo se utilizarán los últimos dos dígitos) ¹	1 para el año y 4 para el correlativo	2 para año y 8 del correlativo (no más de 99,999,999)
Carné (dividido en año y correlativo, en donde para el año sólo se utilizarán los últimos dos dígitos)	1 para el año y 3 para el correlativo	2 para año y 5 del correlativo (no más de 99,999)
Nombre (estimando que no tenga más de 150 caracteres de largo)	50 <i>bytes</i>	50 <i>bytes</i>
Dirección (estimando que no tenga más de 200 caracteres de largo)	100 <i>bytes</i>	100 <i>bytes</i>
Teléfono	4 <i>bytes</i>	9 <i>bytes</i>
Carrera (un número que lo represente)	1	2 (no más de 99 carreras)
Último año en el que se inscribió (los 2 últimos dígitos del año)	1	2
<i>Cantidad total de textos prestados</i>	1	2
<i>Total</i>	166 <i>bytes</i>	182 <i>bytes</i>

- Por cada texto prestado:

¹ ID del carné, ver sección 6.1

Tabla IV. Espacio estimado para almacenar datos de cada texto en la tarjeta

Dato	Formato binario (<i>bytes</i>)	Formato caracter (<i>bytes</i>)
Biblioteca que presta el texto (un número que lo represente)	1	2 (no más de 99)
Persona (bibliotecario) que lo presta (un número que lo represente, suponiendo que no hay más de 9,999 bibliotecarios, en especial, para el formato caracter)	2	4 (si no hay más de 9,999 bibliotecarios)
Texto que se presta (representado por un número, suponiendo que no hay más de 999,999,999,999 textos, en especial, para el formato caracter)	5	12
Fecha límite de devolución (formato aa-mm-dd para el formato carácter, y en formato binario: 2 <i>bytes</i> para el año, 1 para el mes y 1 para el día)	4	6 (año, mes y día tipo char)
<i>Total</i>	12 <i>bytes</i>	24 <i>bytes</i>

Teniendo como una posible política, de que cada estudiante no preste más de 12 textos a la vez dentro de todas las bibliotecas de Ingeniería, se tiene un total de:

- Formato binario (usando la cantidad mínima de *bytes*)
 $12(\text{textos}) * 12 (\text{bytes por texto}) = 144 \text{ bytes}$ para almacenar los 12 textos que pueda prestar cada usuario, más 166 *bytes* de datos generales del usuario, hacen un total de 310 *bytes*.
- Formato caracter (usando la cantidad máxima de *bytes*)
 $12(\text{textos}) * 24 (\text{bytes por texto}) = 288 \text{ bytes}$ para almacenar los textos prestados por cada usuario, más 182 *bytes* de datos generales, hacen un total de 470 *bytes*.

Esta cantidad de *bytes*, representan la cantidad mínima necesaria para almacenar los datos propios de la biblioteca. Si se le agregan funciones de seguridad para el sistema inteligente (como claves de acceso a datos, identificación del usuario por PIN o biometría, etc.), debe agregársele la cantidad necesaria de *bytes*, dependiendo de lo que se desee almacenar, por lo que es recomendable que se busquen dispositivos con más capacidad de almacenamiento que la necesaria (calculada arriba).

Tabla V. Formato de almacenamiento de datos

Dato a almacenar	Formato utilizado para almacenar el dato			
	Formato binario		Formato caracter	
	Formato	Cantidad máxima a representar	Formato	Cantidad máxima a representar
ID del carné – ID emisor (año, últimos dos dígitos)	a	256	aa	99
ID del carné (correlativo)	nnnn	4,294,967,300	nnnnnnnn	99,999,999
Número de carné (año, últimos dos dígitos)	a	256	aa	99
Número de carné (correlativo)	nnn	16,777,216	nnnnn	99,999
Nombre	Una cadena de caracteres tipo char			
Dirección	Una cadena de caracteres tipo char			
Teléfono	nnnn	4,294,967,300	nnn nnnn	999 9999
Carrera	n	256	nn	99
Cantidad de textos prestados	n	256	nn	99
Año de última inscripción	n	256	nn	99
Biblioteca que presta	n	256	nn	99
Bibliotecario que presta	nn	65,536	nnnn	9,999
Texto prestado	nnnnn	1,033,511,600,000	nnn,nnn,nnn,nnn	999,999,999,999
Fecha límite de devolución	a-m-d	256 – 256 – 256	aa-mm-dd	99 – 99 – 99

La tabla V muestra de manera representativa, el formato que se utilizaría para almacenar cada dato dentro de la tarjeta, en donde “n” representa un dígito, “a” es el número que representa el año, “m” el número que representa el mes y “d” el número que representa el día de la fecha límite de devolución, y cada dígito o número representado por las letras “n”, “a”, “m” y “d”, ocupan un *byte* de memoria.

Si se desea el almacenamiento de la huella digital u otros datos dentro de la tarjeta o el botón para aumentar la seguridad del sistema, debe contemplarse la cantidad de *bytes* que utilizaría para almacenarlos, que varía entre 300 *bytes* y 3 *kbyte* (3072 *bytes*), dependiendo del modelo de lector de huellas.

4.2. Análisis de capacidades de cada dispositivo

Existen varios fabricantes de tarjetas inteligentes y cada uno tiene gran variedad de tipos de tarjetas y capacidades. Existen versiones de tarjetas de memoria de contacto, con capacidades que van desde 64 *bits* a 128 *kbits* = 16 *kbytes* (incluso hay de 4 *Mbits*). Las tarjetas con microprocesador de contacto, vienen en versiones que varían entre 256 *bytes* y 64 *kbytes* (aunque ya existen de hasta 512 *kbytes*), pero los más usados varían entre 1 y 4 *kbytes*. Las tarjetas sin contacto, o de radiofrecuencia, normalmente, son más caras y tienen versiones con capacidades que varían entre 16 *bytes* y 2 *kbytes*.

Los botones de memoria tienen versiones con memoria NV RAM que van desde 1 *kbit* hasta 64 *kbits*, las de memoria EEPROM tiene versiones desde 256 *bits* (32 *bytes*) hasta 32 *kbits* (4 *kbytes*).

Para escoger la capacidad adecuada de la tarjeta o el botón, se puede tomar como parámetro, la cantidad de *bytes* que se calcule sean necesarias para que nuestra aplicación funcione de manera correcta y se almacene la información deseada.

También se puede tomar como parámetro, el precio de las tarjetas, la visión que se tenga de ampliar el número de aplicaciones que se puedan manejar por medio del dispositivo inteligente en un futuro, y el grado de seguridad que se desea tener para la aplicación (almacenamiento de PINES, fotos, huellas digitales, etc.).

4.3. Facilidad de uso

Las tarjetas con microprocesador pueden soportar hasta 16 aplicaciones en el mismo *chip*, y la versión DS1991 de *iButton* puede soportar 3 aplicaciones, cada una con su propia clave y área de memoria, lo que facilita el manejo de varias aplicaciones dentro de una sola tarjeta o botón, aplicando un nivel relativamente alto, de seguridad.

Los lectores traen sus propios *drivers*, que permitirán que los lectores sean detectados y adaptados a la máquina en la que se instale, de manera que la configuración del mismo, sea simple.

Ambas tecnologías no son difíciles de usar para el cliente final. La dificultad podría tenerlo al principio el desarrollador de la aplicación, pero no variaría mucho entre una tecnología y otra, y básicamente consistiría en escoger y aprender a usar el lenguaje de programación elegido para desarrollar la aplicación y conocer la estructura del APDU de la tarjeta.

Para la programación de todo lo que se envíe y se reciba desde y hacia el aparato lector para que éste grave o lea la información al dispositivo, se debe de tener conocimientos del estándar que utiliza el fabricante de la tarjeta para realizar estas funciones, puesto que cada fabricante tiene valores específicos para ciertos parámetros del APDU.

Una vez que se cuente con tal información, dependerá de la habilidad del desarrollador de la aplicación para crear el programa que interactúe con las tarjetas inteligentes o los botones.

Usualmente, si se quiere hacer el desarrollo de un sistema por su propia cuenta, se tiene que adquirir el *kit* de desarrollo SDK (*software development kit*) correspondiente (creado por las mismas empresas fabricantes para sus clientes), con el cual viene toda la documentación necesaria para empezar a desarrollar, como lo son las funciones para leer y grabar información con el aparato lector que se vaya a utilizar pero, dependiendo del fabricante, el desarrollador de la aplicación que obtenga esta información debe de comprometerse con el fabricante a mantener la confidencialidad de tal información.

Las tarjetas proporcionan seguridad a un nivel físico, por lo que ya no será necesario preocuparse de esa seguridad. Ahora, a nivel de aplicación, las versiones de lectores de tarjetas con Biometría incorporada, permiten otra opción de seguridad en las tarjetas de manera fácil. Los botones en cambio, no cuentan con lectores que tengan la capacidad de tomar la huella digital, lo que implica que, si se deseara utilizar la huella digital para verificar la identidad de una persona, debe de buscarse por aparte el aparato lector del mismo y la aplicación que lo maneje, cuidando de que sea compatible con el sistema basado en botones.

Tanto los fabricantes de tarjetas inteligentes como los de botones, cuentan con *kits* de desarrollo de *software*, en donde algunos incluyen todo lo necesario para armar un pequeño sistema, con un aparato lector, cables, *drivers* y una pequeña cantidad de tarjetas o botones, para que los clientes puedan experimentar el desarrollo de una aplicación y la facilidad de uso del sistema. Además, en Guatemala ya existen personas que se dedican a la implementación de sistemas basados en tarjetas inteligentes y en botones, aunque los sistemas basados en botones, aún no son tan utilizados porque los sistemas basados en tarjetas son más comunes.

Uno de los aspectos a evaluar a la hora de escoger entre una tarjeta o un botón para un sistema dado es el ambiente en el que se usará, puesto que los botones tienen mayor resistencia a la intemperie y ambientes extremos, a diferencia de las tarjetas.

4.4. Tiempo de vida de los dispositivos

El tiempo de vida de las tarjetas puede verse afectada en gran parte, por la manera como se manipule la misma. El dueño de la tarjeta debe tener un especial cuidado, pues se puede desgastar con facilidad. Se garantiza que el *chip* de la tarjeta, como mínimo, trabajará correctamente para aproximadamente 500,000 inserciones o un lapso no mayor de 5 años.

La duración de la impresión que se haga sobre la tarjeta, de modo que éste se vea bien, depende de la frecuencia de uso de la tarjeta y del tipo de impresión que se le haga. Si es hecha por el fabricante, puede durar hasta 10 años, si es realizada por una impresora diseñada especialmente para este propósito, duraría aproximadamente 1 año, pero si se le pone una capa protectora, puede durar 5 años.

El tipo de capa protectora depende del fabricante, del modelo de la impresora y de las necesidades del cliente. Todas las cintas de color tienen incluida una capa protectora básica, pero existe un *film* o cinta transparente que necesita de una impresora especial y puede ser encargado con hologramas, tal y como el que se utiliza para las licencias de conducir en Guatemala, que tienen hologramas del escudo guatemalteco.

Los botones en cambio, tienen un tiempo de vida mayor, de aproximadamente 10 años equivalentes a más de 10 millones de escrituras, debido al material con el que es fabricado, además es resistente a caídas, rasguños e incluso es resistente al agua y a condiciones tan extremas como el estar dentro de una lavadora.

Los botones, siendo dispositivos de pequeño tamaño, se pueden perder fácilmente, por lo que necesitan de algún otro dispositivo especial para que el dueño de la misma lo cargue, como un reloj, un llavero o un anillo diseñado especialmente para ese propósito. Las tarjetas en cambio, se pueden almacenar en una billetera o en la cartera, siendo más fáciles de encontrar, al igual que las tarjetas de crédito.

Los dispositivos sin contacto, tienen una mayor ventaja sobre los dispositivos de contacto, puesto que éstos se desgastan menos al no tener contacto físico directo con el aparato lector, lo que implica un tiempo de vida mayor pero a un costo más alto y con una gama de versiones de menor capacidad en memoria.

4.5. Tiempo de vida de los lectores

Depende del trato y el uso que se les dé, así como del fabricante del mismo. Lo normal para los lectores de tarjetas de contacto, es que duren aproximadamente 200,000 inserciones de tarjetas, pero los llamados *heavy-duty* pueden durar mucho más. Para los lectores de radiofrecuencia (RFID), es decir, los lectores de tarjetas sin contacto, el tiempo de vida es mucho más largo, debido a que no hay problemas mecánicos.

El tiempo de vida del componente biométrico de los lectores de tarjeta con huella digital, depende mucho de la tecnología y el ambiente, y puede durar por lo menos dos años para el uso diario de por lo menos 200 personas, y que el sensor no tenga ningún problema.

4.6. Análisis de costos

El precio de cada uno de los dispositivos para un sistema basado en tarjetas inteligentes o botones, es proporcional al volumen de compra.

Los precios son los publicados en distintos sitios de Internet, por lo que se les debe sumar los gastos de envío. Los precios pueden variar dependiendo del modelo y la marca del producto.

Tarjetas inteligentes:

- Tarjetas inteligentes con microprocesador de 4 *kbits* tienen un precio aproximado de \$.1.96 cada una con un mínimo de 999 unidades por compra, de 8 *kbits* \$.4.80, y de 16 *kbits* \$.4.95.
- Tarjetas inteligentes de memoria de 256 *bits* tienen un precio aproximado de \$.4.50 cada una si compran menos de 100 unidades o un costo aproximado de \$.3.01 si se compran entre 1,000 y 4,999 unidades.

- Lectores de tarjetas de contacto tienen un precio que varia entre \$35 y \$50 cada uno, dependiendo del modelo (tipo lector de *floppy disk*, conectado al puerto paralelo o puerto USB del CPU, etc.) y de la marca., con un volumen bajo de compra.
- Los lectores de tarjetas con escáner de huella digital incorporado, tienen un precio aproximado de \$.220.00.
- Teclados con lector de tarjeta incorporado de \$.49.95
- Teclados con lector de tarjeta y escáner de huellas digitales incorporado de \$.49.95.
- Las impresoras tienen precios que van desde \$.1,000.00 y \$.6,000.00 dependiendo del modelo, el tipo de impresión (una cara o doble cara), la capacidad que pueda tener de agregarle a la tarjeta un laminado de seguridad, la capacidad de agregarle a la impresora un dispositivo para la codificación de cinta magnética (dispositivo que puede costar \$200.00) y de agregarle un contacto para grabar información en la tarjeta (dispositivo de \$250.00).
- Suministros para impresoras: Los cartuchos de color para 200 impresiones de una sola cara \$.70.75, cartuchos de color para unas 350 impresiones de una sola cara \$.122.75, cartuchos de color negro para 1,000 impresiones de una sola cara \$.10.50, cartuchos de color negro para 1,500 impresiones de una sola cara \$.14.25, cartuchos *full color* con capa protectora para 200 impresiones de \$.93.58, y cartuchos negros con capa protectora para 500 impresiones de \$71.98.
- *Kit* de limpieza de \$.56.75 y rodillos de limpieza (paquete de 5) de \$.30.25 (es necesaria la limpieza de las impresoras cada 10,000 impresiones)
- No necesita de cables especiales para conectar el lector
- El *kit* de desarrollo de ACS tiene un costo de \$180.00 y el *kit* que cuenta con lectura de huella digital tiene un costo de \$450.00.

Figura 17. Impresora para tarjetas



Fuente: http://CardLogix.com/specification/specification_eltronp310_1.html

Botones (precios publicados por Dallas Semiconductor en su página de Internet, a los que hay que agregarle los gastos de envío e impuestos):

- Botones de Dallas Semiconductor de modelo DS1996 con 64 *kbits* (8 *kbytes*) de memoria NV RAM tienen un precio de \$5.73 cada uno. El modelo DS1973 con 4 *kbits* de EEPROM tiene un precio de \$2.18. El modelo DS1993 con 4 *kbits* de NV RAM tiene un precio de \$3.84. El modelo DS1977 con 32 *kbytes* de memoria EEPROM tiene un precio de \$6.89. El precio de todos los modelos es válido si se compran más de 1,000 unidades.
- Lectores de botones *Blue Dot Receptor* DS1402D-DR8 (puerto serial o USB) Y DS1402D-DB8 (puerto paralelo) tienen un precio de \$4.51. Los lectores modelo DS9092, DS9092GT y DS9092T tienen un precio de \$4.24, \$14.37 y \$7.13 respectivamente.
- *Kit iButton Starter Kit* a \$45.00, que incluye un botón de memoria ROM, un botón de 256 *bits* de EEPROM, 1 botón de 4 *kbit* para tomar la hora en tiempo real y 1 de 64 *kbit* de memoria NV RAM, un lector *Blue Dot Receptor* modelo DS1402D-DR8 y un adaptador de puerto serial. Se puede descargar del sitio de *iButton* documentación y código de ejemplos.
- Dispositivos para cargar los botones fabricados por *iButton*: Llaveros de \$14.00 en adelante, anillos y relojes de \$50.00 en adelante, billeteras de \$18.00 en adelante y clips de múltiples usos de \$0.40 en adelante.

El costo a incurrir por cada estudiante tiene que ver directamente con el precio de la tarjeta o botón que se use, la impresión y capa protectora para la tarjeta y el dispositivo para cargar el botón. El costo por biblioteca tiene que ver con el precio de los lectores y los cables necesarios para conectarlos. El costo para el departamento encargado de emitir los carnés, tienen que ver con las impresoras de tarjetas y sus suministros, el aparato lector y sus cables.

El costo inicial para implementar el sistema, depende del precio y licencias del *software* necesario para desarrollar la aplicación, el costo del *kit* de desarrollo para las tarjetas o botones (que incluye la documentación necesaria, los lectores y dispositivos necesarios para conectarlos y unas tarjetas o botones). Además, se debe de tomar en cuenta para el costo inicial, el servicio de los programadores o desarrolladores de la aplicación, en el caso de que se opte por programar la aplicación por si mismos, de lo contrario, el costo inicial para implementar el sistema es el que indique la empresa que proporcione el servicio de *outsourcing*.

Se debe decidir si la primera emisión de carnés, se hace por medio del propio centro emisor (que implicaría la compra de una grabadora de carnés de gran escala debido a la gran cantidad de carnés que se deben de emitir al principio), o dejar que el primer tiraje de carnés se realice contratando *outsourcing*.

Se estima que para empezar, si el sistema está formado por las 4 bibliotecas de la Facultad de Ingeniería, se necesitaría como mínimo para cada una, en base a la cantidad de usuarios que acuden a cada biblioteca, la siguiente cantidad de aparatos lectores, haciendo la salvedad que como mínimo esas bibliotecas deben de contar con esa cantidad de computadoras y se deben de tomar en cuenta los dispositivos necesarios para conectarlos a la computadora, como los cables:

- Biblioteca USAC-UTRECHT: 1 aparato lector
- Biblioteca CICON: 1 aparato lector

- Biblioteca ERIS: 1 aparato lector
- Biblioteca MCC: 2 aparatos lectores

La tabla VI muestra el listado de costos por estudiante, costos por terminal o PC con lectores y otros costos a tomarse en cuenta para iniciar la implementación del sistema inteligente, el cual, se basa en los precios de los dispositivos tomando en cuenta los siguientes aspectos:

- Precios por volúmenes de tarjetas y botones mayores o iguales a 1,000
- Impresión de 1,200 carnés, usando suministro tipo cartucho *full color* con capa protectora para 300 impresiones de \$.122.75 (lo que significa un costo de impresión aproximado de \$.0.41 por tarjeta).
- Lector de tarjeta *CHIPDRIVE micro 110*, con conexión a puerto USB, con capacidad para leer y escribir en cualquier tarjeta de microprocesador, ideal para aplicaciones de uso masivo de tarjetas.
- Lector de tarjeta con biometría (huella digital) modelo *Precise 100 SC* para puerto USB.
- Lector de botones *Blue Dot*
- Adaptador para puerto serial o paralelo.
- Impresora modelo P420 de *CardLogix* con capacidad para imprimir a color en ambos lados de la tarjeta y con opción a agregarle el dispositivo para inicializar el *chip* al momento en el que se imprime.
- Tomar en cuenta gastos de envío e impuestos puesto que los precios presentados, son los precios del distribuidor en Estados Unidos.
- Debido a las diferencias entre las presentaciones de capacidades de las tarjetas y de los botones, se toma como referencia la capacidad de 8 *kbytes*, común en ambos dispositivos, para fines de análisis de costos.

Tabla VI. Listado de costos

Costo por estudiante				
Tarjeta de 8 <i>kbytes</i>	\$ 4.30	Botón de 8 <i>kbytes</i>	\$ 5.73	
Impresión	\$ 0.47	Llavero	\$ 0.56	
Total	\$ 4.77	Total	\$ 6.29	
Costo por lectora en biblioteca				
	normal	con huella		
Lector	\$49.00	\$199.00	Lector	\$ 4.51
			Adaptador de puerto serial	\$12.69
Total	\$49.00	\$199.00	Total	\$ 17.20
Otros costos iniciales para la implementación				
	normal	con huella		
<i>Kit</i> de desarrollo ACS	\$180.00	\$450.00	<i>Kit</i> de iniciación	\$45.00
Total	\$180.00	\$450.00	Total	\$45.00
Costos para la entidad emisora				
Impresora	\$4,090.00			
Dispositivo para inicializar el <i>chip</i> al imprimir	\$250.00			
Cartuchos (4u.)	\$491.00			
<i>Kit</i> de limpieza y rodillos	\$87.00			
Total	\$4918.00			

Fuentes: **Didya_Com - Smart Card Solutions_ Smart Card Reader (s) and more**, www.didya.com; **CardLogix - Smart Cards for Transaction, Identification, and Security Markets**, www.CardLogix.com; **Advanced Card Systems Ltd**, www.acs.com.hk; **Smartcard Focus Home**, www.smartcardfocus.com; **Maxim-Dallas Main Home Page**, store.iButton.com.

4.7. Conclusión

Podemos ver que tanto las tarjetas inteligentes como los botones, tienen versiones con capacidades de almacenamiento grandes, de 64 *kbytes* o más y actualmente se están creando dispositivos con más capacidades, por lo que el tipo de uso que se le darán o el ambiente sobre el que se utilizarán, el grado de seguridad requerido para la aplicación y la cantidad promedio de lecturas/escrituras que se le hagan, serán los factores que determinen cuál dispositivo es más conveniente para un sistema.

Tanto las tarjetas como los botones pueden ser utilizadas para almacenar datos personales y claves personales o de acceso, o pueden ser usadas como monederos electrónicos, pero los botones tienen versiones que rastrean la temperatura, tienen un reloj de tiempo real que puede ser programado, por ejemplo, para tener una fecha de expiración, y son más aptos en los casos en que se deben guardar cualquier tipo de información sobre cualquier objeto, como calibraciones o mantenimientos. Básicamente, los botones son más indicados en los casos en los que los objetos deben llevar información sobre diferentes aspectos en ambientes extremos, los datos cambian frecuentemente, se requiere de durabilidad y es posible que tenga contacto con objetos extraños al lector.

Como se mencionó anteriormente, los botones soportan mayor cantidad de lecturas y escrituras que las tarjetas inteligentes, pues estas últimas, se desgastan con el uso. Los botones se pueden perder fácilmente y si alguien lo encuentra, no podrá saber quién es el dueño del mismo a menos que cuente con un lector, las tarjetas en cambio, con sólo imprimir su nombre y foto en ella, se podrá identificar al dueño de la misma.

Con respecto a los costos, un sistema basado en tarjetas puede resultar más caro a corto plazo que un sistema basado en botones, sin embargo, a largo plazo pueden invertirse los resultados, ya que el costo de un carné de tarjeta por usuario es menor que el costo del carné de botón, y si anualmente se requieren 2,000 carnés, la diferencia entre tarjetas y botones sobrepasa los \$3,000.

Para propósitos de préstamo de textos en las bibliotecas, en donde existen varios usuarios que tendrán que portar su carné con su foto, y luego de evaluar los costos y las características de los sistemas basados en tarjetas y en botones, se recomienda el uso de un sistema basado en tarjetas, utilizando tarjetas con microprocesador con capacidad de 4 *kbytes*. Si se deseara tener otra aplicación en el carné, habría que evaluar el espacio necesario para él y luego considerar la capacidad adecuada.

Si la organización no cuenta con personal de desarrollo de aplicaciones, con los conocimientos suficientes para producir los carnés y crear la nueva aplicación que los maneje de forma eficiente, es recomendable que se contrate un proveedor (servicio *outsourcing*) para que se encarguen de ello, ya que es posible que, por falta de experiencia y conocimientos sobre esta tecnología, surja desorganización en el desarrollo y la integración de los sistemas, y el proyecto resulte ser ineficiente o, en el peor de los casos, sea un fracaso.

Es necesario crear un plan de preparación para los usuarios en donde se definan los procedimientos para preparar y distribuir las tarjetas inteligentes y para reemplazar las tarjetas perdidas, robadas o averiadas.

5. DISEÑO DE LA ESTRUCTURA DE DATOS DENTRO DEL DISPOSITIVO

5.1. Métodos de seguridad

Al hablar de seguridad, podemos identificar tres áreas que debemos tomar en cuenta al implementar este sistema:

- La seguridad de que los datos no sean alterados por terceros
- La seguridad de que los datos del carné no se pierdan
- La seguridad de que el dueño del carné sea el que realice el préstamo

5.1.1. Seguridad de que los datos no sean alterados por terceros

Como se mencionó en el capítulo 2 de este trabajo, existen 2 tipos de tarjetas inteligentes: las de memoria y las de microprocesador.

La diferencia entre ellas la notamos al poner el factor seguridad contra el factor flexibilidad a los cambios, ya que para tener mayor seguridad se sacrifica la flexibilidad de poder agregar más funciones o aplicaciones a la tarjeta en un futuro, de manera más fácil, pero si se quiere tener mayor flexibilidad a los cambios de manera que en un futuro sea más fácil agregarle más funciones a la tarjeta (siempre y cuando los permisos que den las claves, lo permitan), se sacrifica la seguridad, a continuación se explica por qué.

Con las tarjetas de memoria, la seguridad de la información almacenada recae sobre el *software* que maneja las tarjetas, es decir, la aplicación que haga uso de las tarjetas inteligentes, es la que debe de hacer todas las verificaciones necesarias para asegurar que la terminal o PC que accese a la información, sea realmente, la que tiene el permiso de realizarlo, por lo que estas tarjetas son ideales cuando los requerimientos de seguridad permiten el uso de tarjetas con baja o mediana seguridad.

Las tarjetas de memoria, proporcionan gran flexibilidad al poder cambiar la estructura o forma en el que se almacenan los datos, cuando el desarrollador desee agregarle más funciones a la tarjeta. Las tarjetas con microprocesador en cambio, no ofrecen esta flexibilidad, puesto que, una vez se determine la estructura de los datos en la tarjeta, ésta no se podrá modificar para agregarle más funciones a menos que exista el espacio y el permiso para ello.

Esta desventaja de poca flexibilidad, es compensada por las tarjetas de microprocesador, con el grado de seguridad que ofrecen, puesto que, una tarjeta podrá manejar varias aplicaciones con su propia clave de acceso, en donde cada aplicación tendrá su propio directorio, al cual, solamente el *software* definido para la aplicación con la clave correcta, podrá tener acceso. Esto quiere decir que la responsabilidad de la seguridad sobre la información almacenada, recae completamente sobre la tarjeta, pues la propia tarjeta, verificará que la clave de la aplicación (del *software*) concuerde con la clave que está dentro de la tarjeta en el directorio que se le asignó para esa aplicación. Por ejemplo, si una tarjeta tiene como funciones, el manejo de préstamos en biblioteca y el control de acceso al parqueo, el *software* de biblioteca, de ninguna manera tendrá acceso ni borrará datos del directorio asignado para el parqueo y el *software* del parqueo tampoco tendrá acceso al directorio de biblioteca, ya que la misma tarjeta no lo permitirá.

Por todo lo anterior, la elección del tipo de tarjeta a usar, dependerá de la preferencia entre el grado de seguridad que se necesite sobre la información y el grado de facilidad para agregarle más funciones a la tarjeta en un futuro. Se recomienda que se prevean las posibles funciones que pueda tener la tarjeta en un futuro, aunque en un inicio no se implementen todas, para determinar la estructura y la seguridad necesarias dentro de la tarjeta.

Si se decidiera implementar la seguridad a nivel de *software* (que la seguridad no sea manejada por la tarjeta), se puede programar el *software* de manera que cada biblioteca lea únicamente los datos generales del estudiante y los datos sobre los textos que esa biblioteca le prestó, por medio del ID de la biblioteca, que se debe de almacenar en la tarjeta, para cada texto que se preste.

En general, los datos de especial cuidado, que se almacenen dentro de una tarjeta como las claves de acceso o *passwords*, no es necesario que se encripten o codifiquen antes de almacenarse, puesto que las claves utilizadas para proteger la información contenida en su directorio, son suficientes para asegurarla, aunque si se desea, se pueden encriptar pero esto haría más lento el acceso a los datos.

5.1.2. Seguridad de que los datos del carné no se pierdan

Esto es en especial, para los casos en los que el usuario pierda o dañe su carné inteligente, y se necesite recuperar los datos para proporcionarle uno nuevo.

Para que los datos del carné no se pierdan, es necesario que cada biblioteca tenga una pequeña base de datos que contenga un listado con el carné del estudiante y el ID o MFN del texto prestado, de manera que la entidad encargada de emitir los carnés sepa si el usuario está solvente o no en todas las bibliotecas.

En este listado, el almacenamiento de la fecha de préstamo será opcional, puesto que de cualquier forma, el usuario debe de devolver todos los textos que prestó, antes de que se le extienda un nuevo carné.

Cuando un estudiante, solicite su nuevo carné a la entidad emisora (se recomienda que exista una entidad dedicada a proporcionar los carnés de biblioteca), ya sea porque jamás había tenido uno o porque extravió o dañó el que tenía, se extenderá una solicitud a cada biblioteca que participe de este sistema, pidiendo la solvencia del usuario, para que este vea si puede o no entregarle un carné nuevo.

Para esto, podría pedirse al encargado de cada biblioteca que saque un reporte (proporcionado por la misma aplicación que maneje los préstamos) del usuario para saber si está solvente o no y se lo dé para que lo lleve a la entidad emisora, o bien, una persona designada para este propósito por parte de la entidad emisora, podría tener la tarea de visitar cada biblioteca y sacar personalmente tales reportes al finalizar la jornada, todo esto, debido a que actualmente no se cuenta con una red entre todas las bibliotecas, pero como una medida que se aconseja que se tome en un futuro, se recomienda implementar una red entre las bibliotecas, de manera que, la entidad emisora pueda acceder en modo de consulta a todas las bases de datos de todas las bibliotecas para sacar la solvencia de los estudiantes, sin necesidad de molestar a los encargados de cada biblioteca, lo que lo convertiría en un sistema más confiable y óptimo.

Además, como en todo sistema basado en la tecnología de las computadoras, se debe tener el cuidado de sacar *backups* o copias de seguridad de los datos que sean importantes para los usuarios del computador, como lo son los datos de los préstamos realizados, con cierta frecuencia para evitar que, al perder la base de datos ya sea porque falle el disco o por alguna otra razón, no se pierdan totalmente los datos.

5.1.3. La seguridad de que el dueño del carné sea el que realice el préstamo

Este tipo de seguridad se puede implementar de distintas maneras, dependiendo del grado de seguridad que se quiera. Si se requiere de una seguridad baja, se puede imprimir la foto y/o la firma del usuario sobre la tarjeta y que el bibliotecario lo compare con el usuario cuando este llegue a pedir el texto.

Si se requiere de una seguridad media o alta, se puede hacer uso de los pines o claves de acceso, en donde, el pin del usuario debe de estar almacenado dentro del carné en un directorio de sólo lectura, de manera que cada vez que el usuario quiera prestar un texto, se le pida que ingrese su número de pin luego de que el carné haga contacto con el lector, de manera que se verifique que el pin que se ingresó con el pin que esté almacenado dentro del carné, sean los mismos, asegurando así que el portador del carné es el dueño del mismo.

Si se requiere de una seguridad aún más alta, se puede implementar el sistema con el uso de la biometría, por medio de lectores de huella digital o alguna otra característica de las personas. Con las huellas digitales por ejemplo, el portador deberá de colocar su dedo sobre el sensor o lector de huellas, el programa se encargará de compararlo con el que esté almacenado dentro de la tarjeta, para asegurar que el portador es el dueño del carné.

Este tipo de seguridad, también es útil para asegurar que la persona que ingrese al sistema inteligente sea un bibliotecario autorizado y que los datos de la BDD no sean alterados por personas que no estén autorizadas, al proporcionarles a ellos también, un carné con las claves y los permisos necesarios para ello, y configurar el sistema de manera que haga las verificaciones pertinentes antes de permitir que ingrese al sistema o antes de realizar cualquier cambio a la base de datos.

5.2. Estandarización de la estructura de almacenamiento de datos dentro del dispositivo

Como se mencionó en la sección 5.1, al utilizar tarjetas con microprocesador es recomendable que antes de especificar la estructura de almacenamiento o mapa de la tarjeta, se piense un poco acerca de la visión que se tiene en el futuro de expandir las funciones o aplicaciones que pueda manejar la tarjeta, puesto que al grabar el mapa de la tarjeta, se debe de especificar cuántos *bytes* son necesarios para cada directorio y dentro de cada directorio, se debe de especificar cuántos *bytes* tiene cada archivo, teniendo en cuenta de que cada directorio es usado para una función o una aplicación distinta, que hace que el manejo de la seguridad sea más simple al tener un grupo de claves de acceso distinto para cada directorio.

Ejemplos de otros servicios que se podrían manejar dentro de la tarjeta son:

- Pago de retrasadas
- Cursos de vacaciones
- Inscripciones
- Fotocopias
- Parqueo
- Congresos estudiantiles (asistencias)
- Uso de laboratorios (asistencias)
- Impresiones
- Control de créditos, etc.

Si aún no se tiene claro qué aplicaciones se tendrían en un futuro, se puede dejar en blanco o sin definir la porción de la tarjeta que no es necesario usar para la aplicación de la biblioteca, es decir, a la hora de grabar el mapa de la tarjeta, indicarle solamente los *bytes* del directorio que contendrá los datos de la aplicación de biblioteca.

Si se piensa que en un futuro se quieren almacenar otros datos acerca de los préstamos de textos, se puede especificar que el directorio de bibliotecas sea más grande del necesario, ya que, siempre y cuando haya espacio dentro del directorio y se conozca la clave maestra con la que se tiene acceso al mapa de la tarjeta para modificarlo, se pueden agregar archivos a un directorio existente o agregar nuevos directorios a nivel de la raíz.

Si se desea, se pueden estructurar los archivos de manera que cada archivo represente los textos que presta cada biblioteca, es decir, que solamente la biblioteca “x” tenga acceso de modificación a los datos del archivo especificado para ella y que otras bibliotecas tengan permiso de acceso de sólo lectura a tal archivo, evitando que otra biblioteca altere los datos de ella.

Según los datos necesarios a almacenar y el tamaño calculado para almacenar los mismos (ver sección 4.1), se tendrán 2 secciones diferenciadas dentro del dispositivo:

a. Los datos generales del usuario:

- ID del carné
- Carné
- Nombre
- Dirección
- Teléfono
- Carrera
- Cantidad de textos que tiene prestados
- Número del año en el realizó su última inscripción

b. Detalle de textos prestados con la siguiente información:

- Biblioteca que presta el texto
- Persona (bibliotecario) que lo presta
- Texto que se presta
- Fecha límite de devolución

Será necesario que se tenga una pequeña base de datos, en donde por cada biblioteca se tenga el listado de bibliotecarios que trabajan allí, con un número de identificación que será un número correlativo. El texto se identificará con el ID que se obtenga de la base de datos o el número MFN que Winisis utiliza para identificar, internamente, a los textos. No se utiliza el número de clasificación de los textos para identificarlos dentro de la tarjeta, debido a que, como se dijo en el capítulo anterior, la cantidad de caracteres que se utilizan puede ser demasiado grande (puede llegar a tener más de 10 caracteres y si el texto perteneciera a una colección de varios volúmenes o tomos y tuviera varias copias, aumenta aún más la longitud), y esto no optimizaría el uso del espacio dentro del dispositivo.

También se debe de tener un listado de las bibliotecas que utilicen este sistema de préstamos, un listado de las facultades y un listado de carreras que existen en la Universidad. Todos estos listados nos permitirán tener un correlativo para cada dato, este correlativo será el identificador único que va a diferenciar un dato de otro, dentro del mismo listado (como el número de cédula que diferencian una persona de otra).

Tabla VII. Estructura lógica de los datos dentro de la tarjeta

ID del carné – ID emisor			
Número de carné			
Nombre			
Dirección			
Teléfono			
ID Carrera			
Año de última inscripción			
Cantidad de textos prestados en total			
ID biblioteca	ID bibliotecario	ID texto	Fecha límite de devolución

La tabla VIII muestra un ejemplo de la estructura de directorios y archivos dentro de la tarjeta (en formato binario) si se define un directorio para la entidad emisora (con el ID del Carné – ID Emisor), un directorio para los datos del usuario (número de carné, carrera, último año inscrito) y su huella digital, y uno para la aplicación de bibliotecas, teniendo en un solo archivo todos los textos pertenecientes a todas las bibliotecas que haya prestado el usuario y un archivo que indique el total de textos prestados. Además, se deja un espacio sin definir para posibles cambios en el futuro.

Tabla VIII. Ejemplo de estructura de directorios y archivos

Aplicación	Directorio	Tamaño	Archivo	Para	Tamaño (bytes)
Entidad Emisora	A	8 bytes	A1	ID del carné	5 para ID + 3 bytes para futuros cambios
General	B	160	B1	Datos del usuario	160
		“x”	B2	Huella digital	“x”
Bibliotecas	D	145 bytes	D1	Cantidad de textos prestados en total	1
			D2	Detalle de textos prestados	144
Otros directorios	Directorio “E”	“y” bytes	Detalle de archivos para los demás directorios		
Sin definir			Espacio que puede utilizarse para posibles cambios en el futuro		

La tabla IX muestra la estructura de directorios y archivos recomendada para almacenar los datos mínimos que se deben de tener para mantener el control de los préstamos en las bibliotecas, en donde se hace uso de un archivo para cada biblioteca, facilitando así, el control del acceso a los datos evitando que personas o aplicaciones no autorizadas, realicen cambios a los datos ajenos. Por la misma razón, el nombre y el carné se guardan en un archivo de sólo lectura (éstos no cambiaran nunca) y la dirección, el teléfono, las carreras y años de última inscripción en un archivo para lectura y escritura, si es el departamento de Registro y Estadística quien está leyendo el archivo, de lo contrario, será de sólo lectura.

Se puede observar que se dejan 2 *bytes* adicionales, para almacenar el código de otras carreras que esté llevando el estudiante (otras dos carreras simultáneas, 1 *byte* para cada carrera adicional) y los campos que indican cuál es el último año para el que está inscrito en esa(s) carrera(s). Lo anterior, evitará que los estudiantes de carreras simultáneas tengan dos carnés o más.

Tabla IX. Estructura de directorios y archivos recomendada utilizando un archivo para cada biblioteca

Aplicación	Directorio	Tamaño	Archivo	Para	Tamaño (<i>bytes</i>)
Entidad emisora	A	5 <i>bytes</i> de ID + 10 <i>bytes</i> adicionales para posibles cambios en el futuro	A1	ID identificador único de carnés	1 <i>byte</i> para el año + 4 para el correlativo = 5 <i>bytes</i>
			Sin definir, para posibles cambios		10
Datos del usuario	B	160 + 4 <i>bytes</i> adicionales para las carreras simultáneas	B1	Carné	4
				Nombre	50
			B2	Dirección	100
				Teléfono	4
				Carrera principal	1
				Otras carreras (simultáneas)*	2
				Inscrito carrera principal	1
				Inscrito carreras simultáneas *	2
Biblioteca	C	145 + 50 <i>bytes</i> adicionales para posibles cambios futuros	C1	Cantidad de textos prestados en total	1
			C2	Textos MCC	36
			C3	Textos CICON	36
			C4	Textos ERIS	36
			C5	Textos UTRECHT	36
			Sin definir para posibles cambios		50
Sin definir			Espacio sin definir dentro de la tarjeta para posibles cambios en el futuro		

* Para el caso en el que tengan carreras simultáneas

6. DISEÑO DE LA APLICACIÓN

6.1. Descripción general de los sistemas actuales y de lo que sería nuestro sistema inteligente

Los sistemas actuales dentro de las cuatro bibliotecas de la Facultad de Ingeniería, hacen uso del programa Winisis, que es una versión de CDS-ISIS, para almacenar los datos de los textos que tienen disponibles.

El software *CDS - ISIS* (*Computerized Documentation System - Integrated Set for Information System*), comúnmente denominado MicroIisis, fue desarrollado por la UNESCO y es un sistema de almacenamiento y recuperación de información, diseñado para el registro de documentos, permite construir y administrar bases de datos estructuradas, no numéricas, constituidas principalmente por textos. Su uso está dirigido principalmente al área bibliográfica, sin embargo, es aplicable al registro de documentos personales, historias clínicas, información jurídica, expedientes, documentación de proyectos en general y a toda base de datos que posea información estructurada en fichas con datos textuales.

Winisis tiene una interfase gráfica para el sistema operativo *Windows*, y permite, entre otras características, la manipulación simultánea de un número ilimitado de bases de datos, cada una de las cuales puede tener información totalmente diferente que se puede interrelacionar entre sí, permitiendo un máximo de 16 millones de registros dentro de un límite de 500 Mb.

Básicamente, las bibliotecas utilizan Winisis como una base de datos que les permite realizar búsquedas de textos por autor, tema, etc., el programa les muestra todas las coincidencias que obtienen, en base al parámetro de búsqueda que le den, mostrando en pantalla todos los datos bibliográficos de los textos, como su número de clasificación, el nombre del texto y del autor, y todas las características generales de los textos, que hayan ingresado sobre los mismos con anterioridad, por lo que los bibliotecarios utilizan Winisis como una herramienta de búsqueda de datos de textos de su biblioteca, que puede sustituir de alguna manera, la búsqueda de los mismos en los ficheros.

Solamente la Biblioteca MCC cuenta con una aplicación que sirve como interfaz entre el usuario y la base de datos Winisis, que fue desarrollado en *Visual Basic*. Las otras tres bibliotecas utilizan la misma interfaz que proporciona Winisis.

El sistema inteligente que se propone en este trabajo, vendría a complementar el área de control de préstamos, es decir, se ocuparía exclusivamente de ver quién tiene prestado textos de cada biblioteca, ver si tiene moras en otras bibliotecas y proporcionar los datos del usuario, de manera automática al sistema, de modo que en el momento que se desee, se pueda conocer la solvencia de un estudiante o un usuario del personal administrativo para fines como determinar si otro préstamo es posible o si el usuario tiene derecho a inscribirse en el siguiente año, por ejemplo, sin correr un riesgo alto de fraude.

Nuestro sistema funcionará específicamente para leer y grabar datos en el carné inteligente, obtener los datos generales de los usuarios, determinar si el usuario llena los requisitos mínimos para prestar un texto y para dar mantenimiento a la base de datos propia de nuestro sistema. En nuestra aplicación el bibliotecario indicará quién y qué libro presta o devuelve para grabarlos en el carné inteligente, pero este sistema, no se utilizará para ingresar los datos bibliográficos de los textos a la base de datos de Winisis.

Si la biblioteca cuenta con una aplicación propia que maneje los préstamos, es necesario que se adapte tal aplicación para que interactúe con el sistema inteligente, es decir, se debe de programar la interfaz que llame a nuestro sistema y le envíe el número de carné de quien presta o devuelve y el número de clasificación, el ID o el MFN (número interno de registro de Winisis) del texto prestado / devuelto, cada vez que en tal aplicación se indique que un usuario está prestando o devolviendo un texto para que el sistema inteligente actualice los datos en el carné.

Para implementar el sistema inteligente, tendremos una pequeña base de datos en cada biblioteca, y que podemos dividirla en dos partes, la primera parte de esa base de datos será igual para todas las bibliotecas que utilicen esta aplicación, es decir, contendrá los mismos datos en todas las bibliotecas. Esta base de datos común, tendrá un listado con todas las facultades y carreras de la universidad y un listado de todas las bibliotecas que usen esta aplicación y estén autorizados para prestar textos; todos estos listados, estarán debidamente numerados de manera correlativa (ningún número se repetirá dentro de cada listado), de modo que represente el ID (número de identificación) de la facultad, carrera o biblioteca, dentro de nuestra aplicación. Esta porción de la base de datos, también tiene que formar parte de la base de datos de la entidad emisora y otras entidades que necesiten consultar los datos dentro de la tarjeta, como Registro y Estadística que es la entidad en la que los estudiantes se inscriben anualmente, para que pueda actualizar el año de inscripción o la carrera del estudiante, en el caso de que haya cambiado de carrera.

Este ID será el que se almacene dentro de la tarjeta para indicarnos a qué carrera pertenece el estudiante (y por ende, a qué facultad) y en qué biblioteca prestó el texto, de manera que, al leer la tarjeta, tomemos el ID y lo comparemos (a nivel de *software*) con el listado que tenemos en nuestra base de datos, para saber a qué representa ese ID.

La segunda parte de la base de datos será el que maneje en sí, el préstamo de los textos de cada biblioteca y no será igual al de las demás bibliotecas puesto que cada una de ellas almacenará los datos de sus usuarios y los préstamos que se realicen en su biblioteca. Esta parte de la base de datos, tendrá un listado de los bibliotecarios que trabajan en esa biblioteca y sus datos.

También se tendrá un listado de los usuarios que llegan a prestar textos en esa biblioteca con su número de carné, sus datos generales y la(s) carrera(s) a la(s) que pertenece. Esta lista irá creciendo cada vez que la aplicación vea que los datos de la persona que presta los textos, no están dentro de la base de datos, de manera que, solamente aquellas personas que tienen textos prestados o aquellas que frecuentemente prestan textos, estén dentro de esa lista. Para evitar que este listado crezca demasiado y contenga a usuarios que desde hace un tiempo considerable, ya no prestan textos en esa biblioteca, se puede tener a nivel de *software*, una política que indique que cada vez que un usuario devuelva un texto, se revise si aún tiene algún texto prestado de esta biblioteca, si ya no tiene textos prestados (está solvente), entonces se proceda a borrar su registro del listado de usuarios, si no está solvente, que no modifique nada del listado de usuarios.

La segunda parte de la base de datos también contará con un listado de todos los préstamos realizados por los usuarios, indicando el carné del usuario, el texto que presta, el bibliotecario que se lo presta, la fecha de préstamo y la fecha límite de devolución.

Es importante que la entidad emisora tenga su propia base de datos, en la cual, se registren los datos de los usuarios y se lleve el control de los carnés entregados. Se debe de identificar cada carné con un ID propio de la entidad emisora (al que se le dió el nombre de ID de carné – ID Emisor en el capítulo 4), que puede ser formado por los dos últimos dígitos del año en el que se extiende el carné y un número correlativo, para identificar de manera única el carné.

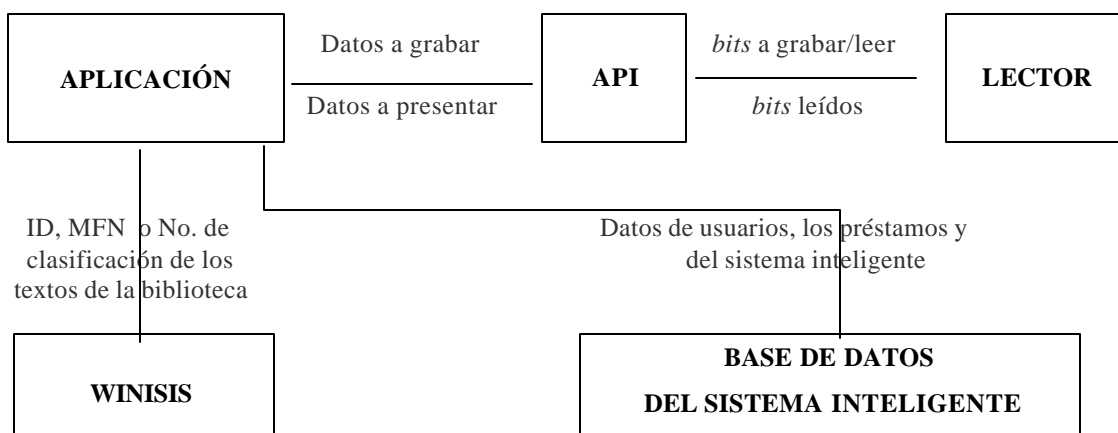
El ID de carné nos servirá para el caso en que los usuarios pierdan su carné, la entidad emisora registraría ese carné con ese ID como un carné perdido o no válido en un listado (o tabla) de carnés inválidos, de manera que a la hora de que un usuario intente prestar un texto, cada biblioteca tenga la capacidad de verificar que el carné insertado aún sea válido o no.

De este modo, se evitará que un usuario tenga dos carnés de biblioteca a la vez, si por ejemplo, un usuario pide un nuevo carné porque lo perdió y luego lo encuentra e intenta prestar textos con ambos carnés, al verificar que uno de ellos no es válido, el bibliotecario podrá retirarle ese carné inmediatamente y tomar las medidas necesarias, haciendo del sistema inteligente, un sistema aún más confiable y óptimo. El listado de carnés que se invaliden, debe de ser distribuido y actualizado en cada biblioteca, ya sea, al finalizar el día por medio de un archivo que pueda ser importado desde la aplicación o, si ya cuentan con una red que comunique todas las bibliotecas con la entidad emisora, realizarlo de manera automática a través de la red.

6.2. Flujo de datos

Básicamente, el sistema estará estructurado de la siguiente manera:

Figura 18. Flujo de datos del sistema inteligente



Aplicación. Es el programa con el que interactúa el bibliotecario para indicar que se le está prestando un texto a alguien o que lo está devolviendo. Aquí se le indicará al sistema quién y qué se está prestando o devolviendo, se encargará de pedir los datos de la tarjeta y pedir que se actualicen los datos tanto en la tarjeta como en la base de datos. Además, verifica si el usuario cumple los requisitos mínimos para poder prestarle un texto y contará con la posibilidad de generar reportes de usuarios morosos.

API. Es el encargado de que el lector y la aplicación se entiendan y puede ser OCF, MUSCLE o PC/SC. Para lograrlo, el API tiene un conjunto de funciones que permiten que la aplicación le diga al lector qué es lo que desea realizar, y el API se encarga de “traducir” estas funciones en llamadas específicas del *driver* del lector que se esté usando, lo que permite que el mismo programa pueda funcionar con distintos tipos de lectores.

Lector. Es el dispositivo físico que lee y graba datos en la tarjeta inteligente o el botón.

Winisis. Sistema que actualmente tienen las cuatro bibliotecas, que constituye una base de datos de todos los textos disponibles en cada biblioteca. De esta base de datos se obtendrá el MFN de un texto a partir de su número de clasificación, o viceversa. Si en un futuro, se decidiera cambiar la base de datos de textos a otra base de datos como *Oracle* o *SQL Server*, se debería de tomar un número correlativo como el ID del texto, de manera que ese ID realice la función que realiza el MFN dentro de nuestro sistema inteligente, que es, identificar un texto de manera única dentro de nuestra base de datos sin usar el número de clasificación, el cual es muy largo.

Base de datos del sistema inteligente. Base de datos que contendrá información general para el control de usuarios y bibliotecarios, e información básica sobre los préstamos realizados en cada biblioteca.

6.3. Descripción del flujo de operaciones en nuestro sistema

a. Para prestar un texto

La aplicación permitirá que se ingrese el número de carné del usuario y el número de clasificación o MFN del texto y llamará a la rutina de préstamo del sistema inteligente.

La rutina de préstamo hará la petición de lectura de la tarjeta, verificará la validez de la misma, y leerá los datos generales del estudiante y los datos de los textos que tiene prestados, buscará dentro de la base de datos del sistema inteligente su número de carné dentro del listado de usuarios que tiene y si ese usuario no existe en el sistema, lo agrega junto con su nombre, dirección y teléfono, y luego verificará si el usuario tiene derecho a más préstamos (esto lo hace al verificar el número de préstamos que tiene el usuario, la carrera a la que pertenece y si está inscrito en el presente año, además, presentando un listado de los textos que tenga prestados y se han pasado de su fecha límite de devolución, datos que se encuentran dentro de la tarjeta). Si no tiene derecho a préstamos, el sistema inteligente presentará un mensaje de alerta, pero si lo tiene, la rutina deberá de enviarle al API todos los datos que se deben de actualizar dentro de la tarjeta (carné del usuario, ID de texto, ID del bibliotecario que lo presta y la fecha límite de devolución) y luego actualizará el listado de préstamos en la base de datos del sistema inteligente.

b. Para devolver un texto

La aplicación permitirá que se ingrese el número de carné del usuario y el número de clasificación o MFN del texto y llamará a la rutina de devolución del sistema inteligente.

La rutina de devolución enviará el número de carné ingresado al API para que éste verifique que el carné insertado en el lector corresponda al usuario que devuelve el texto, verificando antes, que sea un carné válido para la aplicación (que no sea un carné invalidado por robo o extravío) y le envía también el ID del texto para actualizar el carné. Luego, borrará del listado de préstamos de la base de datos del sistema, el registro del texto que fue devuelto y verificará si el usuario aún tiene préstamos pendientes en esta biblioteca, si no los tiene, elimina su registro del listado de usuarios (si se decidiera manejarlo así), si aún tiene préstamos pendientes, no realiza ningún cambio en el listado de usuarios.

6.4. Compatibilidad con otros sistemas

Si ya existiera una aplicación que maneje los préstamos, es necesario que éstas tengan la capacidad de hacer llamadas a rutinas fuera de ellas, si poseen ésta capacidad, se les debe de programar un módulo o rutina que realice las llamadas a los módulos de préstamos o devoluciones del sistema inteligente y les envíe el carné y el número de clasificación o MFN del texto prestado y el ID de la biblioteca que presta el texto.

Además, es necesario que se pueda tener acceso a los datos de las bases de datos que se utilicen, es decir, la aplicación del sistema inteligente debe de tener la posibilidad de obtener información de la base de datos (en especial, información sobre el número de clasificación de los textos), ya que la base de datos del sistema inteligente se basará en la base de datos de textos existente (en nuestro caso, Winisis) para que los números de clasificación de los textos concuerden con el sistema utilizado en cada biblioteca en particular.

En otros términos, la base de datos de la aplicación existente debería de almacenar como mínimo, los mismos datos que se almacenan en lo que en este trabajo se llamó como la segunda parte de la base de datos del sistema inteligente (es decir, los listados de bibliotecarios, usuarios y préstamos) y proporcionar el ID del texto prestado a partir del número de clasificación (función que aquí se obtiene con Winisis), y la aplicación debería de ser capaz de ser programada de manera que haga las llamadas a los módulos que interactúan con la tarjeta y realizar las verificaciones necesarias para los préstamos. Además, se deberá de programar que la información a actualizar cada vez que se preste o devuelva un texto, sea sobre la base de datos de la aplicación existente.

Una alternativa a la necesidad de que se pueda acceder a la base de datos de la aplicación ya existente, sería el poder exportar un listado de los textos ingresados en esa base de datos y se importe a nuestro sistema, guardando en una tabla (listado) de textos, los datos del MFN, el título y número de clasificación de los textos, de modo que, el sistema inteligente pueda realizar las verificaciones necesarias para saber si el número MFN de los textos ingresados, es válido (existe en Winisis) o no.

También es importante tomar en cuenta que el API y el lenguaje de programación que se escoja para la implementación del sistema puedan manejar los *drivers* del lector que se elija.

6.5. Módulos a programar

6.5.1. Lectura

El módulo de lectura, es el que se encarga de obtener todos los datos contenidos en el carné inteligente, le pide al API que verifique la tarjeta y las claves en ella, y almacena los datos en variables globales (en memoria) de nuestra aplicación para que cualquier módulo tenga acceso a la información leída.

Recibirá como parámetro desde el módulo que lo llame, el número de carné del usuario y retornará un valor de éxito, fracaso o cancelación en la operación de lectura.

1. Recibe como parámetro el carné del usuario.
2. Verificar si el carné está insertado en el lector, si no lo está, presentar un mensaje para que el bibliotecario decida entre continuar con la operación o salir de ella. Si decide salirse, retornar un valor de cancelación de lectura.
3. Pedir al API que verifique que la tarjeta es válida para esta aplicación (bibliotecas) y si es válido en el sistema (no está registrado como carné robado o extraviado), continuar si es válido, si no, presentar un mensaje indicando el error ocurrido y salir del procedimiento retornando un valor de fracaso en la operación de lectura.
4. Pedir al API que verifique si el PIN que ingresó o la huella digital del usuario, coincidan con el que esté almacenado en la tarjeta (esto solamente si se implementara con estas funciones de seguridad), continuar si es válido, si no, presentar un mensaje indicando el error ocurrido y salir del procedimiento retornando un valor de fracaso en la operación de lectura.
5. Leer los datos del usuario: Último año de inscripción, nombre, dirección, teléfono y carrera(s), almacenándolos en variables globales.
6. Verificar que el número de carné recibido por parámetro coincida con el carné insertado, si no es así, presentar un mensaje indicando el error ocurrido y salir del procedimiento retornando un valor de fracaso en la operación.

7. Leer la cantidad de textos que tiene prestados y almacenarlo en una variable global, si la cantidad de textos prestados es mayor al máximo permitido por carné (en el capítulo 4 se definió 12 textos como máximo por carné, cantidad que puede ser cambiada según lo definan los bibliotecarios), asignarle a la variable global que controle si puede o no puede prestar textos, el valor de **no puede** prestar más textos. Si el valor de la cantidad de textos prestados es mayor a 0, continuar con el paso 8, si no, ir al paso 9.

8. Sacar un listado (leer) de todos los textos que tiene prestados el usuario y almacenarlos en una variable global (tipo *recordset* o un arreglo, por ejemplo). Mientras saque este listado, ir contando la cantidad de textos que pertenecen a **esta** biblioteca, si tiene la cantidad máxima de textos prestados permitidos en **esta** biblioteca, asignarle a la variable global que controle si puede o no puede prestar textos, un valor de **no puede aquí** prestar más textos. También, al ir sacando el listado de textos, deberá de ir sacando un listado de textos que hayan pasado de la fecha límite de devolución, almacenándolo en una variable global (un *recordset* o arreglo), si tiene textos con mora, asignarle a una variable global que controla si es o no moroso, el valor de **es moroso**.

9. Retornar un valor de éxito en la operación de lectura. (En este momento, las variables globales, ya contendrán toda la información del usuario).

6.5.2. Desencriptar

Luego de la investigación realizada, se ha determinado que no es necesario encriptar los datos que se almacenen dentro de la tarjeta, debido a que la información contenida en ella, está bien protegida por las características propias de la tarjeta, por lo que no se considera necesario un módulo para desencriptar.

6.5.3. Procesar

Este módulo es el que maneja todo el proceso para prestar o devolver un texto, desde pedir los datos del usuario y el texto que se presta / devuelve, pedir que se lea el carné, pedir que se actualice el carné, hasta actualizar la BDD de préstamos del sistema inteligente. El módulo de procesar, puede dividirse en 2 submódulos: Cuando se presta un texto y cuando se devuelve un texto.

a. Préstamo

1. Inicializar valor de todas las variables globales.
2. Ingreso de número de carné del usuario y número de clasificación del texto a prestar y la cantidad de días que se le va a prestar (tendrá un valor por defecto que pueda modificarse).
3. Buscar en la base de datos Winisis (o en el listado de textos de nuestro sistema, si así se manejara), el ID o MFN del texto que tiene ese número de clasificación, si no existe en la base, enviar un mensaje a la pantalla (posible error al ingresar el número de clasificación) pidiendo que se vuelva a ingresar el número de clasificación y dar la opción de salir de la operación. Si reingresa el número de clasificación, que vuelva a revisarse si existe ese texto en Winisis, si sí, continuar con el paso 4.
4. Leer la tarjeta (usar el módulo de lectura del punto 6.5.1) enviándole como parámetro el número de carné.
5. Si ocurrió algún error en la operación de lectura, salir del procedimiento, si no, continuar.

6. Si el usuario está inscrito en el año en curso, continuar, si no, presentar el mensaje de que no está inscrito y salir del procedimiento.
7. Si tiene derecho a prestar textos porque no ha rebasado el límite de préstamos por carné ni ha rebasado el límite de préstamos en la biblioteca actual (se verifica si el valor de la variable global tiene el valor de **puede**, de **no puede**, o de **no puede aquí**). Si el valor fuera **no puede**, se presenta el mensaje de que ha rebasado la cantidad de préstamos de textos permitidos por carné, si fuera **no puede aquí**, se presenta el mensaje de que ha rebasado la cantidad de préstamos de textos permitidos en esta biblioteca, y si fuera **puede**, continuar con el paso 8.
8. Verificar el valor de la variable global que controla si el usuario es moroso. Si su valor fuera **es moroso**, se presenta el mensaje de que tiene moras, y se presenta el listado de textos morosos, indicando a qué biblioteca pertenecen y la fecha en que tuvo que ser devuelta, y se le da la opción al bibliotecario de continuar con el proceso de préstamo (es decir, ignorar que debe textos en otras bibliotecas) o cancelar la operación. Si acepta continuar, seguir con el paso 9, de lo contrario, salir del procedimiento. Si el valor de la variable fuera **no es moroso**, continuar con el paso 9.
9. Agregar el registro del texto que se va a prestar (dentro de la variable global que contenga los textos prestados por el usuario) y actualizar la variable global que controla la cantidad de textos prestados por el estudiante (sumarle un texto).
10. Buscar en la base de datos del sistema, dentro del listado de usuarios, el registro del usuario que tiene ese carné, si no existe, agregar su carné, nombre, dirección y teléfono al listado (insertar en la tabla de usuarios).

11. Actualizar la tarjeta usando el módulo grabar del punto 6.5.5.
12. Actualizar en la BDD, el listado de préstamos (insertar en la tabla de préstamos, el carné del usuario, el ID del texto prestado, el ID del bibliotecario que realiza la operación de préstamo, la fecha de préstamo y la fecha límite de devolución).

b. Devolución

1. Inicializar valor de todas las variables globales.
2. Ingreso de número de carné del usuario y número de clasificación del texto a devolver.
3. Buscar en la base de datos Winisis, el ID o MFN del texto que tiene ese número de clasificación, si no existe en la base, enviar un mensaje a la pantalla (posible error al ingresar el número de clasificación) pidiendo que se vuelva a ingresar el número de clasificación y dar la opción de salir de la operación. Si reingresa el número de clasificación, que vuelva a revisarse si existe ese texto en Winisis, si sí, continuar con el paso 4.
4. Leer la tarjeta (usar el módulo de lectura del punto 6.5.1) enviándole como parámetro el número de carné.
5. Si ocurrió algún error en la operación de lectura, salir del procedimiento, si no, continuar.
6. Verificar si ese texto está en mora o no, si sí presentar el mensaje, presentar una pantalla para cobrarle el costo de la mora y continuar.

7. Quitar del listado de textos, el texto que se va a devolver (dentro de la variable global que contenga los textos prestados por el usuario) y actualizar la variable global que controla la cantidad de textos prestados por el estudiante (restarle un texto).
8. Actualizar la tarjeta usando el módulo de grabar del punto 6.5.5.
9. Actualizar la BDD, el listado de préstamos (borrar o hacer un *delete* en la tabla de préstamos, al registro con el carné del usuario y el ID del texto ingresados).
10. Verificar si el usuario aún tiene textos prestados de esta biblioteca (*select* de todos los registros dentro de la tabla de préstamos, a todos aquellos registros cuyo carné sea igual al ingresado), si no existe registro alguno, borrar el registro del usuario, del listado de usuarios (*delete* en la tabla de usuarios, al registro con el carné del usuario ingresado). Esto con el fin de evitar que la base de datos crezca demasiado y únicamente si así lo desearan los bibliotecarios, es decir, si a ellos no les interesara guardar información de usuarios que no les deben texto alguno.

6.5.4. Encriptar

Como se mencionó en el punto 6.5.2, luego de la investigación realizada, se ha determinado que no es necesario encriptar los datos debido a que las claves de acceso en las tarjetas de microprocesador, realizan un buen trabajo para proteger la información contenida en la tarjeta, por lo que no es necesario un módulo para encriptar.

6.5.5. Grabar

El módulo de grabar, es el que se encarga de empaquetar los datos a actualizar sobre la tarjeta (forma el APDU) y se lo envía al API para que le indique al lector que actualice los datos. Toda la información, la toma de las variables globales que han sido manipuladas en los módulos de lectura y procesar.

1. En base a los datos contenidos en la variable (*recordset* o arreglo) global que contiene los textos prestados por el usuario, formar el APDU (empaquetar los datos indicando ID de texto, biblioteca, bibliotecario y fecha límite de devolución) siguiendo los protocolos definidos para la tarjeta/lector que se haya seleccionado.
2. Pedir al API que grave los datos contenidos en el APDU.
3. En base al valor contenido en la variable global que controla la cantidad de textos prestados por el usuario, formar el APDU que actualice la cantidad de textos prestados en la tarjeta
4. Pedir al API que grave los datos contenidos en el APDU.
5. Retornar un valor de éxito de operación.

6.5.6. Volver a hacer tarjeta

Volver a hacer la tarjeta es lo mismo que grabar el mapa de la estructura de un carné nuevo (hacer un nuevo carné). Con el fin de no entregar un carné nuevo a un usuario que no está solvente con las bibliotecas, se debe proceder a:

1. Verificar en todas las bibliotecas que el usuario esté solvente (por medio de una búsqueda en el listado de préstamos de cada biblioteca).
2. Si el estudiante no está solvente, debe de devolver todos los textos o pagar su costo a la biblioteca que se lo prestó, antes de que se le pueda extender su nuevo carné.

6.5.7. Otros módulos

Otros módulos que deben programarse son:

1. **Reporte de usuarios.** Reporte en el que se consulta la base de datos del sistema inteligente para obtener el listado de todos los usuarios a los que la biblioteca ha prestado textos (listado de la base de datos en las bibliotecas) u obtener el listado de usuarios a los que se les ha entregado un carné (listado de la base de datos en la entidad emisora). Hace un *select* a todos los registros de la tabla de usuarios.
2. **Reporte de préstamos.** Reporte con el listado de todos los textos prestados por un usuario, podrá ser utilizado para determinar si se le puede extender o no un carné nuevo a un usuario o para cualquier otro fin que tengan en las bibliotecas. Hace un *select* a todos los registros de la tabla de préstamos, cuyo carné sea igual al ingresado en pantalla.
3. **Reporte de morosos.** Este reporte, les servirá a los bibliotecarios para sacar el listado de todos los usuarios que estén morosos, con todos sus datos, para poder contactarlo. Hace un *select* a todos los registros de la tabla de préstamos, cuya fecha límite de devolución sea menor a la fecha actual o la ingresada en pantalla.

4. **Módulo de mantenimiento o ABC de bibliotecarios.** Módulo que permitirá registrar todas las altas, bajas y cambios de los datos de los bibliotecarios, dentro de la base de datos del sistema inteligente.

5. **Módulo de devoluciones especiales.** Este módulo se utilizará para cuando los usuarios devuelvan los textos sin usar su carné inteligente por la pérdida del mismo, o tengan que pagar el costo de los textos que deben a las bibliotecas, y será utilizado únicamente para que el usuario quede solvente en la biblioteca y pueda pedir su carné nuevo o para el caso en el que el usuario haya perdido el texto prestado. Este módulo deberá de borrar el registro de los textos que devuelva o pague el usuario en la base de datos en el listado de préstamos (un *delete* al registro del texto que devuelva o pague en la tabla de préstamos), y luego de verificar que el usuario ya no debe ningún texto (verificando con un *select* a todo registro de la tabla de préstamos que tenga ese carné y no devuelva registro alguno), borrar su registro del listado de usuarios (un *delete* al registro del usuario en la tabla de usuarios).

6. **Módulo de actualización de datos.** Este módulo es programado especialmente para la entidad encargada de inscribir a los estudiantes. Con éste módulo, se tendrá la capacidad de acceder al directorio, dentro del carné, que contiene los datos del usuario, para actualizar el último año de inscripción, su carrera (en caso de que cambie), su dirección y teléfono.

7. **Módulo de mantenimiento de carreras y facultades.** Módulo para modificar los datos de las carreras y facultades de la universidad, dentro de la base de datos del sistema inteligente.

8. **Módulo de mantenimiento de bibliotecas.** Módulo para modificar o agregar los datos de las bibliotecas que formen parte de las bibliotecas que usen el sistema inteligente, dentro de la base de datos del sistema inteligente.
9. **Módulo de invalidación de carnés.** Módulo por medio del cual, la entidad emisora invalidará los carnés que se reporten como perdidos o robados. Hace un *insert* en una tabla de carnés inválidos, almacenando el ID de carné de los carnés que se invaliden.
10. **Consulta de validez.** Con este módulo se verificará en red si el carné se reportó como perdido o robado (no válido). Este módulo deberá de ser llamado por el módulo de procesar (punto 6.5.3), como una verificación antes de actualizar los datos del préstamo en el carné, si el carné no fuera válido se debe de presentar el mensaje correspondiente y salir del módulo procesar.
11. **Inicializar variables.** Encargado de darle el valor inicial a las variables globales. Todas las variables de tipo entero se inicializan con un valor 0, las de tipo *string* (cadena) se inicializan con cadenas en blanco, las de tipo *recordset* o arreglo se inicializan con todos sus campos en blanco o con valor 0 (dependiendo del tipo de dato del campo) y las variables de estado (como el que indica si se puede prestar o es moroso) se inicializan con el valor que indique que puede prestar, no es moroso y que no ha ocurrido ningún error.

6.6. Módulos para cada biblioteca

Las bibliotecas deberán de contar con los módulos descritos del punto 6.5.1 al 6.5.5 y del punto 6.5.7, los módulos del 1 al 5, el 11, 7 y 8 (estos dos últimos se usan sólo si la entidad emisora, encargada de la gestión y administración del sistema, ingresa nuevas facultades, carreras o bibliotecas), y el módulo de consulta de validez (el 10) consultando una tabla especial (de carnés invalidados) en la base de datos de la biblioteca si no tiene red, o consultando las bases de datos de la entidad emisora, si estuviera en red con la entidad emisora.

El administrador de la aplicación en la biblioteca (o bien, el encargado principal de la biblioteca), será quien maneje de manera exclusiva, los módulos de mantenimiento de bibliotecarios, carreras y facultades, y bibliotecas, y el módulo de devoluciones especiales. Los demás módulos, podrán ser utilizados por los bibliotecarios.

La entidad emisora contará como mínimo con el módulo 6.5.6 de volver a hacer tarjeta y los módulos del 7 al 10 del punto 6.5.7, y si estuviera en red con las bibliotecas, el módulo de reporte de usuarios y reporte de préstamos (números 1 y 2 del punto 6.5.7) y el de consulta de validez (número 10 del punto 6.5.7) en modo de consulta en red.

La entidad encargada de inscribir a los estudiantes y personal administrativo, deberá de contar con el módulo de actualización de datos (número 6 del punto 6.5.7) y si estuviera en red con las bibliotecas y la entidad emisora, el módulo de reporte de préstamos (número 2 del punto 6.5.7) y el de consulta de validez (número 10 del punto 6.5.7) en modo de consulta en red.

6.7. Lenguajes que se pueden usar para programar la aplicación

La aplicación que interactúe con los bibliotecarios, puede ser programada en cualquier lenguaje, dependiendo del gusto y experiencia del desarrollador, así como también, de la compatibilidad con el API que se seleccione, entre ellos podemos mencionar:

- Delphi
- Borland C++
- Visual C++
- Visual basic
- Oracle developer

CONCLUSIONES

1. El uso de un carné único para todas las bibliotecas que lleve consigo la información actualizada sobre los datos generales del usuario y los textos que tengan prestados, les permite a los bibliotecarios llevar un mejor control sobre los préstamos que realizan, obteniendo de manera automática los datos necesarios para contactar al usuario, conociendo quién es el responsable de cada préstamo y evitando que se le entregue un carné nuevo a personas que tienen moras en las bibliotecas.
2. Los bibliotecarios necesitan evitar que los usuarios presten textos con carnés ajenos, minimizar el tiempo de atención al usuario, facilitar la verificación de los antecedentes de los estudiantes, verificar si los usuarios están o no inscritos en la universidad en el año en curso y saber quién fue el responsable de prestar un texto dado, con el fin de mejorar el servicio de la biblioteca.
3. Las tarjetas inteligentes y los botones son medios de almacenamiento que se están utilizando actualmente en el mundo entero para distintas aplicaciones en las que es necesario almacenar información actualizada de las personas u objetos y se necesita transportarlos de un lado a otro. Ambas tecnologías tienen versiones con propiedades que las hacen más aptas para ciertas aplicaciones que para otras.

4. El sistema de identificación universitaria actual ha llevado a los bibliotecarios a adoptar políticas restrictivas que hacen del servicio de la biblioteca, un servicio poco eficiente puesto que los estudiantes muchas veces no respetan tales políticas y se llevan los textos aunque esté prohibido. El sistema de identificación que se propone, ofrece automatizar el sistema de préstamos, minimizando el tiempo de atención para cada estudiante y el tiempo de verificación de antecedentes de los estudiantes, proporciona los datos para contactar al usuario sin que él mismo llene algún formulario y permite que el bibliotecario decida si se le presta o no un texto al saber, por medio de la información almacenada en el carné inteligente, si el estudiante llena los requisitos necesarios para ello y conocer las posibles moras que tenga en otras bibliotecas que hacen uso del sistema inteligente.

5. Luego de un análisis de las características de las tarjetas y los botones, y de un análisis de costos, se estimó que un sistema basado en tarjetas inteligentes, es el que mejor se adapta a los requerimientos y recursos de las bibliotecas. La tarjeta inteligente le permitirá a los bibliotecarios verificar la identidad del usuario al poder observar la foto de él, impresa sobre la tarjeta, obtener automáticamente desde el carné los datos necesarios para contactarlo en caso de ser necesario, y a la vez les permitirá conocer en cualquier momento, la cantidad y qué textos tiene prestados el usuario, las moras que tenga con otras bibliotecas, y verificar si está inscrito en el presente año, lo que permitirá al bibliotecario tener un mejor control y evitar realizar préstamos a estudiantes morosos o que no llenen los requisitos necesarios. Además, les permitirá manejar el nivel de seguridad necesario para asegurar los datos sobre los préstamos y permitirá que en un futuro, las tarjetas tengan más funciones.

6. Es útil agrupar los datos que pertenezcan a una aplicación en su propio directorio, lo cual evitará que otras aplicaciones alteren sus datos. Además, tener un archivo para cada grupo de textos prestados por cada biblioteca, evitará que las demás bibliotecas sobre-escriban accidentalmente los datos de los textos que prestó en otra biblioteca. Tener un archivo especial de los datos generales del usuario que sea de sólo lectura, evitará también, que estos datos sean borrados por accidente.
7. Es importante que se tenga una base de datos con la información sobre los préstamos realizados, ya que, en caso de pérdida de un carné o a la hora de crear carnés nuevos, se podrá conocer por medio de esa base de datos, si el usuario tiene alguna mora en cualquier biblioteca.
8. La inversión a realizar para implementar un sistema basado en tarjetas inteligentes puede resultar más caro a corto plazo a diferencia de la implementación de un sistema basado en botones, sin embargo, el costo por carné es menor al utilizar tarjetas que al utilizar botones, lo que significa que a largo plazo pueden invertirse los resultados en cuanto a costos. Por otro lado, debido al mismo hecho de que el costo por tarjeta es menor que el costo por botón, el sistema basado en tarjetas resulta ser más accesible para el usuario final que tiene pagar por obtener el carné.
9. En un sistema basado en tarjetas se podrá evitar el anonimato del carné, no así, en un sistema basado en botones, puesto que sobre las tarjetas se puede imprimir la foto y los datos del dueño del mismo, de manera que si se diera el caso en el que se perdiera el carné, cualquier persona podrá saber a quién pertenece el carné.

RECOMENDACIONES

1. Capacitar al personal de las bibliotecas para que conozcan los beneficios del uso de sistemas automatizados, para evitar la resistencia al cambio, puesto que, en el caso del sistema propuesto, el mayor beneficio es poder mantener un mejor control de los préstamos y de los carnés que se extienden a los usuarios, evitándoles pérdidas de tiempo y material bibliográfico. Además, se debe de reforzar la costumbre de ingresar al sistema con la clave de acceso personal al comenzar a usar el sistema inteligente y salir del sistema si se va a retirar, para evitar que terceros presten textos en nombre de otro bibliotecario.
2. Acostumbrarse a sacar copias de seguridad o *backups*, para evitar que se pierdan los datos de las bases de datos por cualquier eventualidad que se presente y apoyar al centro que se encargue de emitir los carnés, proporcionándoles los datos que requieran para verificar la solvencia de los estudiantes antes de que se les entregue un carné nuevo.

BIBLIOGRAFÍA

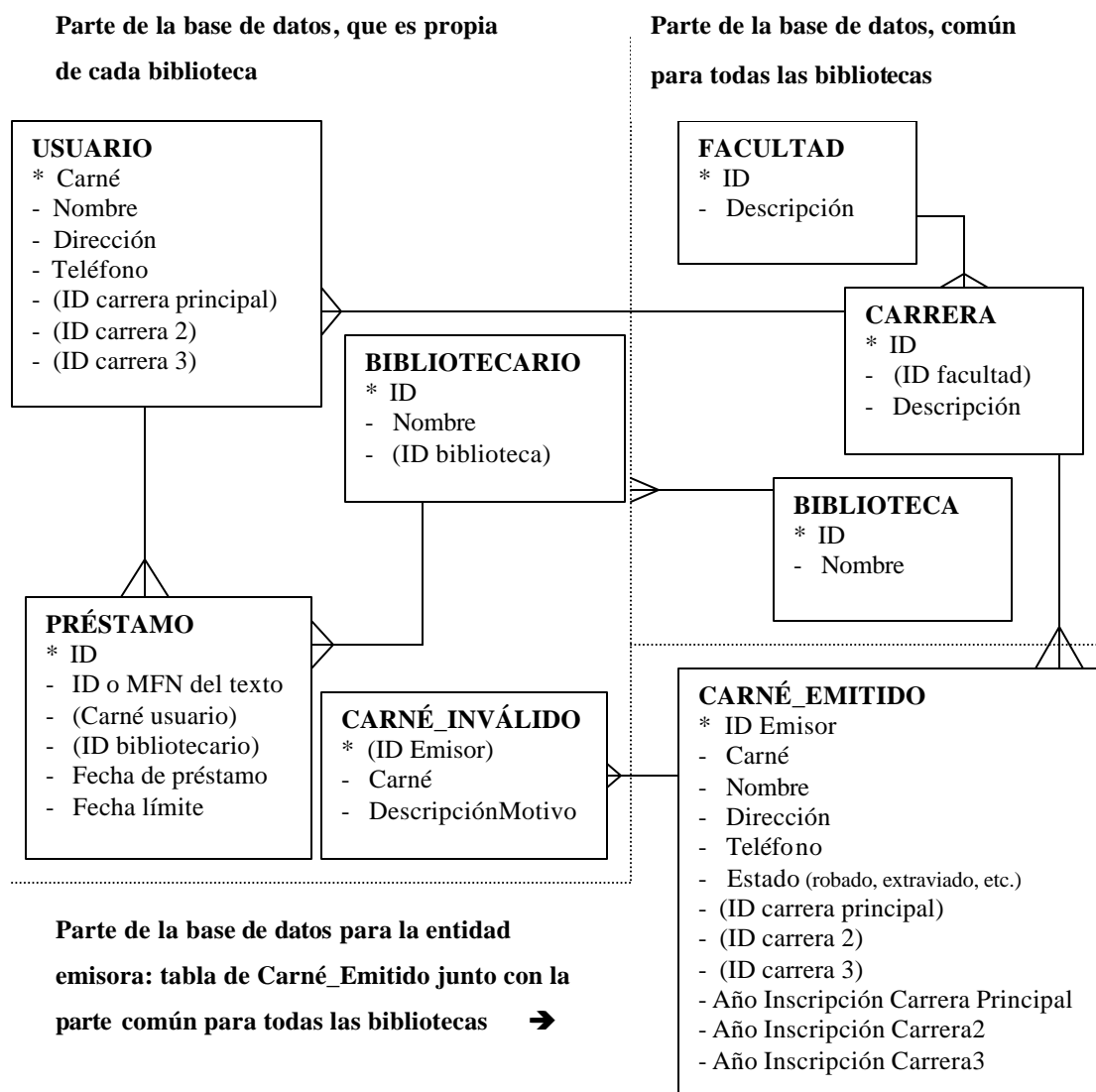
1. http://200.6.42.16:2001/e_isis_dll.html. **ISIS_DLL**. junio 2002.
2. http://ap-electronics.com/tarjetas_inteligentes.asp. **AP-ELECTRONICS_com - Tienda Electrónica - Tarjetas - Programadores - Satelite - Receptores. Tarjetas Inteligentes**. agosto 2001.
3. <http://ca.banesto.es/documentos/tarjetas/>. **CA Banesto - Tarjetas inteligentes**. septiembre 2004.
4. <http://es.tldp.org/Manuales-LuCAS/doc-unixsec/unixsec-html/node112.html>. **Sistemas basados en algo poseído tarjetas inteligentes**. agosto 2004.
5. <http://store.iButton.com>. **Maxim-Dallas Main Home Page**. septiembre 2004.
6. <http://www.csi.map.es/csi/silice/Global29.html>. **Tarjetas inteligentes**. agosto 2001.
7. <http://www.dcc.uchile.cl/~rbaeza/cursos/proyarq/hlopez/inf2.html>. **Tecnología de Smart Card**. agosto 2004.
8. http://www.eluniversal.com.mx/pls/impreso/ol_tu_dinero.html?p_id_seccion=7&p_id_nota=1114. **El Universal Online. Tarjetas: Abran paso al chip**. septiembre 2004.
9. http://www.exceldata.es/espanol2/espanol/aboutsmart_esp.html. **M_MAR Group. Acerca de las tarjetas inteligentes**. septiembre 2004.
10. <http://www.gemplus.com/smart/basics/what.html>. **Smart Technology at Gemplus. What's so smart about smart cards?** septiembre 2004.
11. <http://www.iButton.com/index.html>. **iButton - Contact Memory, Digital Temperature Data Loggers, Java-powered and Secure eCash Tokens**. septiembre 2004.
12. <http://www.microsoft.com/latam/technet/articulos/windows2k/smtcard/>. **Microsoft TechNet -Tarjetas inteligentes**. agosto 2001.
13. <http://www.monografias.com/trabajos10/tarin/tarin.shtml>. **Tarjetas inteligentes**. agosto 2004.

14. <http://www.newtech.com.ve/tarjetas.htm>. **NewTech Solutions Group Tarjetas Inteligentes**. septiembre 2004.
15. <http://www.nortropic.com/gerencia/g319/tarjeta.htm>. **Tarjetas inteligentes: Cuando el futuro nos alcanza**. septiembre 2004.
16. <http://www.semiconductors.philips.com/markets/identification/products/mifare/>. **MIFARE - contactless Smart Card Ics**. agosto 2001.
17. <http://www.todomba.com/displayarticle297.html>. **Tarjetas Inteligentes y su aplicación en los Programas de Fidelización News Todo MBA España - Buscador, ranking, noticias y becas de Masters**. agosto 2004.
18. <http://www.upm.es/informacion/carneupm/infogen.html>. **Las tarjetas inteligentes**. agosto 2004.
19. Schildt, Herbert. **C++ para programadores**. España: McGraw-Hill / Interamericana de España, S.A., 1996.
20. www.acs.com.hk. **Advanced Card Systems Ltd**. septiembre 2004.
21. www.axalto.com. **Axalto Home - Smart card solutions for multi-application security in**. septiembre 2004.
22. www.CardLogix.com. **CardLogix - Smart Cards for Transaction, Identification, and Security Markets**. agosto 2004.
23. www.didya.com. **Smart Card Solutions**. septiembre 2004.
24. www.dte.us.es/tec_inf/itis/peri_intrabajos/TarjetasInteligentes.ppt. **Tarjetas inteligentes**. mayo 2004.
25. www.scia.org/default.htm. **Knowledgebase overview with frams**. agosto 2001.
26. www.sciam.com/0896issue/0896fancher. **Scientific American Article Smart Cards August 1996**. agosto 2001.
27. www.smartcardfocus.com. **Smartcard Focus Home**. septiembre 2004.

APÉNDICE

APÉNDICE 1

Figura 19. Diseño de la base de datos con el mínimo de tablas y campos que debe tener



APÉNDICE 2

Pseudocódigo de los módulos principales del sistema

Variables globales

- a. IDemisor – ID del carné, del emisor, valor inicial = cadena vacía
- b. Carné – carné del usuario que presta o devuelve un texto, valor inicial = 0
- c. Globnombre – nombre del usuario, valor inicial = cadena vacía
- d. Globdireccion – dirección del usuario, valor inicial = cadena vacía
- e. Globtelefono – teléfono del usuario, valor inicial = cadena vacía
- f. Globcarrera – carrera(s) del usuario, arreglo con el siguiente tipo de estructura:
 - Carrera – ID de la carrera, valor inicial = 0
 - Inscrito – último año en el que se inscribió para esa carrera, valor inicial = 0
- g. Texto – número de clasificación del texto a prestar o devolver, valor inicial = 0
- h. Globcontador – índice que apunta al registro o posición en la cual se encuentra el registro que contiene los datos del texto a devolver.
- i. Globcantidad – cantidad de textos prestados por el usuario, valor inicial = 0
- j. Globmoroso – bandera que indica si el usuario es moroso en esta o cualquier otra biblioteca, valor inicial = Nomoroso
- k. Globtextomoroso – bandera que indica si el usuario tiene un texto específico de esta biblioteca en estado moroso, valor inicial = Nomoroso
- l. Globinscrito – bandera que indica si el usuario está inscrito en el presente año, valor inicial = falso
- m. Globprestar – bandera que indica si el usuario puede realizar préstamos, valor inicial = Nopuede
- n. Globestado – estado de la lectura de un carné (Ej. Lectura exitosa, carné inválido, etc.)

- o. Globtextos – arreglo que contendrá el listado de textos prestados por el usuario, arreglo con el siguiente tipo de estructura:
 - Biblioteca – ID de la biblioteca que presta el texto, valor inicial = 0
 - MFN – MFN del texto prestado, valor inicial = 0
 - Bibliotecario – ID del bibliotecario que lo prestó, valor inicial = 0
 - Fechalímite – fecha límite de devolución del texto, valor inicial = vacío
- p. Globmoras – arreglo que contendrá el listado de textos prestados por el usuario, que ya hayan pasado de la fecha límite de devolución, con la misma estructura de la variable Globtextos.
- q. Globfechaactual – fecha actual del sistema, valor inicial = fecha de hoy
- r. Globdíaspréstamo – contiene el valor de días por defecto, que se prestará los textos, valor inicial = al que se decida en cada biblioteca, Ej. 10 días hábiles.
- s. Globfechalímite – fecha límite de devolución de un texto, si se prestara el día de hoy (igual a Globfechaactual + Globdíaspréstamo)
- t. Globbiblioteca – ID de la biblioteca que está prestando el texto, valor inicial = ID de la biblioteca
- u. Continuar – para el módulo de Procesamiento de Préstamos, si encontró alguna mora, guardará S si el bibliotecario desea continuar con el procedimiento de préstamos, N si no.
- v. Globarchivo – nombre del archivo a modificar, que corresponde al archivo de **esta** biblioteca, dentro del carné inteligente.

Procedimientos o funciones principales

1. **Lectura**: leer el carné y validarlo con la aplicación.
2. **Grabar_cambios**: mandar el APDU que contenga los datos a grabar en la tarjeta y grabarlo.
3. **Inicializar_variables**: darle el valor inicial a todas las variables globales mencionadas arriba.

4. **Leer_datos_generales:** leer el nombre, el carné, la dirección, el teléfono y la(s) carrera(s) desde el carné insertado en el lector, y almacenarlos en las variables globales correspondientes. Leer también el último año de inscripción, para la carrera que corresponda, del estudiante, si coincide la inscripción de alguna de las carreras con el año actual, asignarle a la variable global globinscrito un valor de **verdadero**, si no, **falso** (el estudiante está inscrito en el presente año).
5. **Leer_textos_prestados:** lee el listado de textos prestados por el usuario, desde el carné inteligente y lo almacena una variable global (globtextos). Mientras saque este listado, contará la cantidad de textos que pertenecen a **esta** biblioteca, si tiene la cantidad máxima de textos prestados permitidos en **esta** biblioteca, asignarle a la variable global globprestar el valor de **no puede aquí**. También, irá sacando un listado de textos que hayan pasado de la fecha límite de devolución, almacenándolo en una variable global (globmoras), si tiene textos con mora, asignarle a la variable global globmoroso el valor de **es moroso**, si no se le deja el valor de **no moroso**.
6. **Leer_cantidad_prestados:** lee la cantidad de textos prestados por el usuario, desde el carné inteligente, y almacena el valor en la variable globcantidad.
7. **Buscar_usuario_en_BDD:** ver si los datos del usuario están ingresados en la base de datos del sistema inteligente, si no lo están, insertarlos.
8. **EncontrarTexto:** encuentra la posición en la que se encuentra el registro de un texto determinado, dentro del carné inteligente, devolviendo el valor del *byte* o el valor del puntero del registro que contiene el dato.
9. **Insertar_préstamo:** insertar en la tabla de préstamos de la base de datos del sistema inteligente, los datos del nuevo préstamo.

10. **CobrarMora**: módulo para cobrar la multa por mora.
11. **Borrar_préstamo**: borrar de la tabla de préstamos de la base de datos del sistema inteligente, el registro de un préstamo para indicar que ese texto fue devuelto.
12. **ExisteMFN**: busca el número MFN del texto dentro de Winisis, utilizando la funciones de la librería ISIS.DLL de Winisis, si lo encuentra, lo almacena en la variable global **globMFN** y devuelve el valor **true** a la rutina que lo llamó, de lo contrario devuelve un valor **false**. Para ver cuál es el MFN de un texto con determinado número de clasificación, se utilizará la función `IsisSrcSearch` para encontrar el registro que contenga el número de clasificación del texto, al encontrarlo, se utilizará la función `IsisSrcMfnMap` para que nos retorne el número MFN del texto. Ambas funciones, `IsisSrcSearch` e `IsisSrcMfnMap`, son funciones que están especificadas dentro de la librería ISIS.DLL de Winisis.
13. **Borrar_usuario**: luego de que se devuelva un texto, `borrar_usuario` revisará si el usuario aún tiene textos prestados en esta biblioteca, si no los tiene, borra su registro de la tabla de usuarios, si los tiene, no realiza ninguna operación.
14. **Presentar_moras**: presentar en pantalla, los datos contenidos en el arreglo `globmoras` que contiene los textos morosos, presentando por cada texto, el número MFN, el nombre de la biblioteca a la que pertenece el texto y la fecha límite de devolución del texto.
15. **Agregar_texto_al_listado**: recibe por parámetro el número MFN del texto que se presta y la fecha límite para entregarlo, y lo agrega al arreglo `globtextos` junto con el código (ID) de esta biblioteca, el ID del bibliotecario que lo presta, la fecha de préstamo y la fecha límite de devolución del texto.

16. **Borrar_texto_del_listado:** quita del listado globtextos el registro con MFN igual al número MFN que recibe por parámetro, es decir, deja en blanco los campos de ese registro.
17. **Procesar_préstamo:** proceso completo para el préstamo de textos que comprende desde el ingreso del número de carné y MFN del texto a prestar, la lectura del carné, la verificación de moras, hasta la actualización de los cambios tanto en la tarjeta como en la base de datos de la biblioteca.
18. **Procesar_devolución:** proceso completo para la devolución de textos que comprende desde el ingreso del número de carné y MFN del texto a devolver, la lectura del carné, la verificación y cobro de moras, hasta la actualización de los cambios tanto en la tarjeta como en la base de datos de la biblioteca.

Lectura (carné)

```

begin
  if ( carne_insertado is true) then // si se insertó el carné en el lector
    begin
      if(carne_válido is true) then // si el carné es válido para la aplicación de
      bibliotecas
        begin
          leerdatosgenerales( ) // leer datos generales del dueño del carné
          if (id_emisor _válido is true) then // si el carné es válido (tiene un id emisor
válido)
            begin
              if (carné == carneleído) then // si el carné ingresado es igual al carné leído
desde la tarjeta
                begin
                  leercantidadprestados( )
                  leertextosprestados( )
                  return exitolectura
                end else
                  begin
                    write (“el carné insertado no coincide con el número de carné ingresado”)
                    return errorcarnesnocoinciden
                  end
            end
          end
        end
      end
    end
  end

```

```

        end else          // carné insertado no es válido, fue reportado como robado o
extraviado
        begin
            write (“el carné insertado fue invalidado, fue reportado como robado o
extraviado”)
            return errorcarneinvalidorobado-extraviado
        end
    end else // carné insertado no es válido para la aplicación de biblioteca
    begin
        write (“el carné insertado no es válido para la aplicación de biblioteca”)
        return errorcarneinvalido
    end
end else // carné no ha sido insertado aún
begin
    write (“el carné aún no ha sido insertado en el lector”)
    return errorcarnenoinsertado
end
end

```

Grabar_cambios()

```

var
    cadena string
    cont integer
begin
    for (cont = 0, cont < globcant, con++)
    begin
        if (globtextos[cont].biblioteca == estabiblioteca) then
        begin
            cadena = globtextos[cont].carné + globtextos[cont].mfnc +
globtextos[cont].biblioteca +
                globtextos[cont].bibliotecario + globtextos[cont].fechalímite
            formar apdu (cadena)
            enviar apdu al lector // junto con el nombre del archivo a afectar (globarchivo) y
la posición
        end
    end
    cadena = globcantidad
    formar apdu (cadena)
    enviar apdu al lector // junto con el nombre del archivo a afectar y la posición
end

```

Buscarusuario_en_BDD (carné)

var

sql *string*
selección *recordset*

begin

sql ← “select carné
from usuario
where carné = ” + nocarné

selección ← ejecutar (sql)

if (selección == eof) *then*

begin

sql ← “insert into usuario (carné, nombre, dirección, teléfono, id carrera
1,
id carrera 2, id carrera n)
values (” + nocarné + “,” + nombre + “,” + dirección + “,” + telefono +
“,” +
carrera[0] + “,” + carrera[1] + “,” + carrera[2] + “)”
ejecutar (sql)

end

end

Insertar_préstamo (carné, mfn, bibliotecario, fechaprestamo, fechalimite)

begin

var sql *string*

sql ← “insert into prestamo (carnéusuario, idtexto, idbibliotecario, fecha de
préstamo,

fecha límite)
values (” + carné + “,” + mfn + “,” + bibliotecario + “,” + fechaprestamo
+ “,” + fechalimite + “)”

ejecutar (sql)

end

Borrar_préstamo (carné, mfn)

begin

var sql *string*

sql ← “delete
from prestamo
where carnéusuario = ” + carné + “and idtexto = ” + mfn + “)”

```

        ejecutar (sql )
    end
    borrar_usuario ( carné )
    var
        sql string
    begin
        sql ← “delete
        from usuario
        where carné = ” + carné + “
        and not exists (
            select *
            from prestamo
            where carnéusuario = ” + carné + “)”

        ejecutar (sql )
    end

    Encontrar_texto (texto)
    begin
        var contador, moroso, encontro as integer

        contador = 0
        moroso = 0
        encontro = 0

        while ((contador < globcantidad) and (encontro = 0))
            // mientras el contador no exceda de la cantidad de textos almacenados en el carné y
            no se haya
            // encontrado su registro
            begin
                if globtextos[contador].mfñ = texto then
                    begin
                        if globtextos[contador].fechalímite > globfechaactual then
                            begin
                                globtextomoroso = moroso
                            end
                            encontro = 1 // ya se encontró el texto que se desea devolver
                        end else
                            contador = contador + 1
                        end
                    end
                if encontro = 1 then
                    begin
                        return contador // retornar el índice del registro en el que se encuentra el texto
                    end
                end
            end
        end
    end
    buscado

```

```

        end else
            return -1 // valor a retornar si no se encontró el texto
        end
    end
end

Procesar_préstamo ( )
    continuar char
    fechalimite date
begin
    continuar ← “s”
    inicializar_variables( )
    write (“ingrese el no. de carné del usuario”)
    read ( carné )
    write (“ingrese el no. de clasificación del texto a prestar”)
    read ( texto )
    write (“ingrese la cantidad de días a prestarle el texto”)
    read ( globdías )
    globfechalímite ← globfechaactual + globdíaspréstamo
    if (existe_mfn( texto ) is true) then
        begin
            globestado = lectura (carné)
            if (globestado == exitolectura ) then // no ocurrió ningún error en la lectura de
la tarjeta
                begin
                    if (globinscrito == falso ) then
                        write (“no se puede efectuar el préstamo: el usuario no está inscrito”)
                    else
                        begin
                            if (globprestar == nopuedeaquí ) then
                                write (“el usuario ha rebasado la cantidad límite de préstamos de ésta
biblioteca”)
                            else
                                begin
                                    if (globprestar == nopuede) then
                                        write (“el usuario ha rebasado la cantidad límite de préstamos”)
                                    else
                                        begin
                                            if (globmoroso == esmoroso ) then
                                                begin
                                                    write (“el usuario es moroso”)
                                                    presentar_moras( )
                                                    write (“¿desea continuar con el préstamo (s/n)? ”)
                                                    read(continuar)

```

```

        if (continuar == "s") then
        begin
            agregartexto_al_listado(globmf, globfechalímite)
            globcantidad ← globcantidad + 1
            buscarusuario_en_bdd(carné)
            grabar_cambios ( )
            insertar_préstamo(carné, globmf, bibliotecario, fechadehoy,
                            fechalímite)
        end
    end
end
end
end
end
end
end else // el número de clasificación ingresado es erróneo
begin
    write ("el número de clasificación de texto ingresado, no existe en la base de
datos")
    return errortextoingresado
end
end

```

Procesar_devolución ()

var

fechalímite date

begin

inicializar_variables()

bandera ← 0

write ("ingrese el no. de carné del usuario")

read (carné)

write ("ingrese el no. de clasificación del texto a prestar")

read (texto)

if (existemfn(texto) is true) then

begin

globestado = lectura (carné)

if (globestado == exitolectura) then

begin

globcontador = encontrartexto(texto)

if (globcontador = -1) then // no se encontró en el carné, algún registro de
ese texto

begin

write ("el texto ingresado, no ha sido prestado a este usuario")

return errortextonoprestado

end else

```

        if (globtexto[globcontador].fechalímite < globfechaactual) then // tiene
mora
        begin
            write (“este texto debió de haberse devuelto el día ”, fechalímite)
            cobrarmora(texto)
        end
        borrartexto_del_listado(globmfn)
        globcantidad ← globcantidad - 1
        grabar_cambios ( )
        borrar_préstamo(carné, globmfn)
        borrar_usuario(carné)
        end
    end
end else // el número de clasificación ingresado es erróneo
begin
    write (“el número de clasificación del texto ingresado, no existe en la base de
datos”)
    return errortextoingresado
end
end

```