Universidad de las Ciencias Informáticas

Facultad II

Título: Prototipo de aplicación para la gestión de información en Tarjetas Inteligentes Java

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

Autores: Elvis Vázquez Aragón.
Yuri G. Fuentes Lora.

Tutor: Ing. Wilson Eriberto Aguilera Ávila.

28 de junio del 2007
"Año 49 de La Revolución."

"El éxito de los hombres no se mide por su éxito inmediato, sino por su éxito definitivo:- no se mide por el dinero que acumularon, sino por el resultado de sus obras."

José Martí.

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

_____	_____	_____
Firma del Autor	Firma del Autor	Firma del Tutor

DATOS DE CONTACTO

Tutor: Ing. Wilson Eriberto Aguilera Ávila.
Universidad de las ciencias Informáticas, Habana, Cuba.
Email: wilson@uci.cu

AGRADECIMIENTOS

A La Revolución por haber puesto en nuestras manos la posibilidad de formarnos en esta Universidad y de ser partícipe de un sueño de nuestro Comandante.

A nuestro comandante por haber pensado en nosotros como la generación del futuro.

A nuestros padres por su apoyo incondicional por haber inculcado en nosotros los mejores valores y habernos guiado por el camino del conocimiento.

A nuestro tutor por haber trabajado de conjunto con nosotros y por su dedicación y esmero.

A nuestros amigos que han sido la mejor compañía durante estos cinco años y con los que hemos pasado momentos inolvidables.

A todos aquellos que de una forma u otra nos ayudaron en el desarrollo de este trabajo.

DEDICATORIA

A mis padres por confiar en mí y por indicarme siempre el camino correcto a seguir. A mi hermana.

A mi esposa por ser quien es, por darme siempre su apoyo y por soportarme durante tanto tiempo. A mis amigos, especialmente a Emilio por su ayuda. Y a todos aquellos que de una forma u otra hicieron posible este momento.

Yuri Gaspar Fuentes Lora.

A mis padres, por tener confianza en mí y alentarme en los momentos más difíciles. A mi hermano, por ser mi guía en el camino del conocimiento. A mi tío Andrés por exhortarme a ser un mejor profesional. A mi novia por haber sido paciente conmigo y ayudarme cuando más lo necesitaba. A mi abuela, muy en especial por su cariño y ternura. Y a todas aquellas personas que me han brindado su apoyo.

Elvis Vázquez Aragón.

RESUMEN

El presente trabajo tiene como objetivo fundamental el desarrollo de un prototipo de aplicación que gestione información de la historia clínica de un portador en una Tarjeta Inteligente Java (TIJ) haciendo uso de la tecnología JavaCard. Se realizó una investigación acerca de la evolución y desarrollo de la Tarjeta Inteligente (TI), y en específico de la tecnología JavaCard, la cual fue utilizada para desarrollar este prototipo. Esto permitiría evitar el excesivo manejo de papeles, en el momento de recibir atención médica y mejorar la seguridad y la privacidad de la información.

PALABRAS CLAVE

- Tarjetas Inteligentes.
- JavaCard.
- Historia Clínica.

TABLA DE CONTENIDOS

INTRODUCCIÓN

El futuro de los sistemas y de la tecnología de la información en general no podría ser mejor, en particular todo lo vinculado a servicios de salud, a sus usuarios, pacientes y otros protagonistas. Prácticamente todas sus categorías están avanzando, incorporando al sector un procesamiento y una resolución considerable de problemas.

El desarrollo tecnológico de las aplicaciones basadas en Tarjetas Inteligentes ha ido incrementando a nivel mundial. Esta tecnología posee grandes ventajas en cuanto a la gestión de la información, la seguridad y la portabilidad. Aprovechando las ventajas que estas nos ofrecen, es que se decide hacer un estudio para aplicar dicha tecnología en diferentes servicios tales como la salud, para gestionar la información de la Historias Clínicas[1] (HC) y en otras que necesiten manejar la información de forma oportuna y rápida. Las Tarjetas Inteligentes permiten operar desde un simple control de acceso, hasta complejas combinaciones que pueden incluir la información personal del usuario.

Numerosos campos de las aplicaciones de las TI requieren que datos y recursos sean protegidos por altos mecanismos de seguridad.

"El lenguaje de programación Java representa una respuesta práctica a las cuestiones de movilidad y seguridad. Actualmente, se puede afirmar que Java se ha impuesto como un estándar en estos dominios de aplicación donde las exigencias de seguridad son altas." [1]

JavaCard es un nuevo miembro de las tecnologías Java, que está orientada a la programación de tarjetas inteligentes, diseñadas de tal forma, que ciertas construcciones de Java consideradas como demasiado complejas o no aplicables para la programación de las mismas no son incorporadas. Por otro lado, se agregan facilidades específicas para el manejo de transacciones con tarjetas inteligentes. Las políticas de seguridad de Java que prohíben cualquier interacción entre objetos de diferentes applets[2] fueron modificadas, y en algunos casos, debilitadas: JavaCard, por ejemplo, permite que un objeto sea compartido por diferentes applets.

El caso de estudio que se ha considerado consiste en un sistema para almacenar y proteger la información personal y la Historia Clínica de un paciente en una Tarjeta Inteligente.

[1] Historia Clínica: Documento que incluye la identificación personal de un paciente así como los datos referentes a sus enfermedades (diagnóstico y tratamiento).
[2] Los applets JavaCard son las aplicaciones de usuario, que pueden ser descargadas a la tarjeta después de que esta haya sido emitida.

Situación Problémica:

En la actualidad las Historias Clínicas de los pacientes presentan algunos problemas en cuanto al proceso de gestión de la información. En ocasiones la información del paciente se extravía, por lo general el acceso oportuno a la información médica de emergencia es tardío; en caso de asistir a un hospital al cual no pertenezca entonces se tiene que crear una nueva HC a partir de un resumen de la anterior entidad hospitalaria, por lo que se pierde parte de la información que se tenía de su seguimiento médico. En caso de emergencia, las vías rápidas de acceder a la información del paciente son limitadas, por lo que no se conocerían datos de importancia como: grupo sanguíneo, alergias, enfermedades, medicamentos en uso, lo cual pudiera traer complicaciones para la salud del paciente.

Posible solución:

De ahí que surja la idea de informatizar el proceso de gestión de información haciendo uso de Tarjetas Inteligentes, específicamente TIJ, así el paciente tendría en todo momento sus datos clínicos almacenados en una tarjeta, y sería de fácil portabilidad.

¿Que ventajas traería aplicar esta tecnología en el Sistema de Salud?

El paciente portaría su HC en todo momento ya que seria del tamaño de su carne de identidad, se evitaría el excesivo manejo de papeles, en el momento de recibir atención médica, así como también se puede identificar al paciente. Esta tarjeta puede almacenar información médica básica del paciente como tipo de sangre, alergias y medicinas necesarias, en caso de no existir una conexión directa y teniendo un lector de este tipo de tarjetas se puede obtener la información crítica del paciente.

Las tarjetas inteligentes de salud pueden mejorar la seguridad y la privacidad de información de paciente, provee un transportador seguro para registros médicos portátiles, reduce fraude, da soporte a los nuevos procesos para registros médicos portátiles, provee acceso seguro para la información médica de emergencia, provee la plataforma para implementar otras aplicaciones según necesite el Ministerio de Salud (MINSAP).

Estas tecnologías usadas con dispositivos informáticos permiten mantener un historial del paciente, confirmando la elegibilidad del mismo para los servicios, maximizando los beneficios y contraprestaciones, protegiendo los datos confidenciales e incrementando la fidelidad del paciente.

¿Porque aplicar la tecnología de las TI a las Historias Clínicas?

La Historia Clínica es un documento médico el cual contiene información del paciente, esta información debe estar protegida ya que ahí se almacenan los datos personales del paciente así como las enfermedades, tratamiento y diagnostico que ha tenido a lo largo de su vida.

Esta información solamente es accesible por el personal médico autorizado los cuales están obligados a mantener el secreto de la información conocida. El mantenimiento de la confidencialidad y privacidad de los pacientes implica primeramente que la HC, debe estar custodiada de forma adecuada, permaneciendo accesible únicamente al personal autorizado.

Es por esto que se decide aplicar la tecnología de Tarjetas Inteligentes, a las HC, ya que las TI ofrecen la seguridad necesaria para proteger dicho documento, el cual siempre estará en manos del paciente y protegido por un PIN (Personal Identification Number).

Problema Científico:

¿Cómo gestionar información mediante el desarrollo de un prototipo de aplicación, empleando la tecnología de tarjeta inteligente Java e interactuar con un servidor de base de datos?

Objeto de Estudio:

- Proceso de gestión de información referente a la identificación de persona e historia clínica de paciente, en una Tarjeta Inteligente Java.

Objetivo General:

- Desarrollar un prototipo de sistema que intercambie información con una Tarjeta Inteligente Java e interactué con un servidor de base de datos.

Objetivo Específico:

- Desarrollar un prototipo de aplicación que se ejecute sobre un emulador de Tarjeta Inteligente Java, capaz de intercambiar información.
- Desarrollar un prototipo de aplicación que interactué con el cliente y un servidor de base de datos, y que genere información para intercambiar con el emulador de la tecnología de Tarjetas inteligentes Java.

Campo de Acción:

Gestionar la información referente a la identificación de persona e historia clínica de paciente emulando el comportamiento de una Tarjeta Inteligente Java, e interactuar con un servidor de base de datos.

Tareas a Realizar:

- Escoger un lenguaje y una herramienta que permita el desarrollo de aplicaciones en tarjetas inteligentes.
- Modelar el proceso de desarrollo del software para la gestión de información referente a la historia clínica de pacientes en la tarjeta inteligente.

Actualidad y necesidad del trabajo.

El MINSAP dentro de su estrategia de informatización ha incluido el empleo futuro de Tarjetas Inteligentes para almacenar información médica de pacientes, es por ello que es necesario conocer toda la tecnología referente a las Tarjetas inteligentes para poder enfrentar este proyecto.

Aportes prácticos esperados del trabajo.

Se espera un documento de tesis donde se explique de forma clara y detallada el funcionamiento de las actuales tecnologías de Tarjetas Inteligentes (Conceptos, Normas, tipos de tarjetas, Herramientas, funcionamiento, particularidades, lectores, etc.).

Un prototipo de sistema que intercambie información con una Tarjeta Inteligente Java e interactúe con un servidor de base de datos y sea capaz de gestionar la información referente a la identificación de persona e historia clínica de paciente.

Estructuración del contenido con una breve explicación de sus partes.

Capítulo 1: Fundamentación teórica.

Durante este capítulo se proporciona una panorámica general acerca de las Tarjetas Inteligentes; de manera que se revisa la historia de su desarrollo, los tipos de tarjetas que existen actualmente y sus aplicaciones. Además se incluye una explicación breve de sus características físicas y de las funciones que realiza su sistema operativo.

Capítulo 2: Características del Sistema.

En este capítulo se describen la realización de los Casos de Usos, se definen los actores y trabajadores que participan en el negocio, quedan explícitas las reglas del negocio, y se brinda además una representación gráfica del Modelo de negocio. Además se realiza la descripción del sistema a automatizar, se hace un diagrama de casos de uso del sistema para un mejor entendimiento de como se va a informatizar el mismo. Se definen los actores del sistema y los requerimientos que debe cumplir la aplicación de software que se implementó. Se definen los requerimientos mínimos que debe tener el sistema de cómputo donde se vaya a instalar la aplicación.

Capítulo 3: Análisis y diseño del sistema.

Tras la definición y descripción, en el capítulo anterior, de las funcionalidades deseadas y necesarias del sistema propuesto; se hace necesario definir cómo se desarrolla. En este capítulo se realiza el diagrama de clases del diseño dando respuesta a la solución que se propone, se da seguimiento y realización a los casos de uso del sistema. Se describen además los principios de diseño que se tendrán durante el desarrollo de la aplicación.

Capítulo 4: Implementación.

En este capítulo se describe cómo los elementos del modelo de diseño se implementan en términos de componentes, para esto se muestra el diagrama de componentes. además se muestra el diagrama de despliegue con el objetivo de mostrar la distribución física de los nodos de cómputo que necesita la aplicación.

Capitulo 5: Estudio de factibilidad.

En este capítulo se realiza una planificación basada en casos de uso con el objetivo de conocer el costo de desarrollar el software.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción.

Durante este capítulo se proporciona una panorámica general acerca de las Tarjetas Inteligentes; de manera que se revisa la historia de su desarrollo, los tipos de tarjetas que existen actualmente y sus aplicaciones. Además se incluye una explicación breve de sus características físicas y de las funciones que realiza su sistema operativo.

1.2 Tarjeta Inteligente.

1.2.1 Definición.

Se define a la tarjeta inteligente como un dispositivo que posee una apariencia similar a la tradicional tarjeta de crédito, que contiene un pequeño chip[3] incrustado que controla el acceso a la información y brinda protección física a los datos almacenados. (Ver Figura 1.1)

"…el chip puede tener dos funciones, ser un poderoso microprocesador o actuar como un chip de memoria. El chip de silicio tiene tres funciones principales:

1. *Almacenamiento de datos.*
2. *Seguridad en la información.*
3. *Procesamiento de datos".* [2]

La transferencia de datos puede llevarse a cabo a través de los contactos que se encuentran en la superficie de la tarjeta, (Ver Figura 1.3), o sin contactos por medio de campos electromagnéticos. (Ver Figura 1.4).

[3] Un Chip es una pieza de Silicio fusionada con circuitos electrónicos. También se conoce como Circuito Integrado

Figura 1.1: Ejemplo de Tarjeta Inteligente con contactos, una Tarjeta Inteligente esta compuesta por un cuerpo de plástico y un circuito integrado (chip).

1.2.2 Surgimiento.

Las tarjetas inteligentes surgen ante nuevas necesidades del mercado, las cuales no pueden ser satisfechas por la tarjeta de banda magnética. Esta tecnología surge en los 70 cuando inventores de Alemania Juergen Dethloff, Japón Arimura y Francia Moreno inscribieron las patentes originales en 1968, 1970, 1974 respectivamente. Muchos de los trabajos relacionados con las TI, estuvieron en investigación hasta la década de los 80, debido a que la tecnología de los semiconductores no estaba lo suficientemente desarrollada.

Las antecesoras de las TI fueron las Tarjetas Magnéticas (*Magnetic Stripe Cards*), utilizadas en cajeros automáticos y como tarjetas de crédito. Estas almacenan su información en una banda magnética la cual esta adherida a su superficie.

1.2.3 Tipos de tarjetas según sus capacidades.

"Las tarjetas se pueden clasificar según las capacidades de su chip, las tarjetas más habituales son:

- ***Memoria:*** *tarjetas que únicamente son un contenedor de ficheros pero que no albergan aplicaciones ejecutables. Éstas se usan generalmente en aplicaciones de identificación y control de acceso sin altos requisitos de seguridad.*
- ***Micro procesadas****: tarjetas con una estructura análoga a la de un ordenador (procesador, memoria volátil, memoria persistente). Éstas albergan ficheros y aplicaciones y suelen usarse para identificación y pago con monederos electrónicos.*
- ***Criptográficas****: tarjetas micro procesadas avanzadas en las que hay módulos hardware para la ejecución de algoritmos usados en cifrados y firmas digitales. En estas tarjetas se puede almacenar de forma segura un certificado digital (y su clave privada) y firmar documentos o*

autenticarse con la tarjeta sin que el certificado salga de la tarjeta (sin que se instale en el almacén de certificados de un navegador Web, por ejemplo) ya que es el procesador de la propia tarjeta el que realiza la firma. Un ejemplo de estas tarjetas son las emitidas por la Fábrica Nacional de Moneda y Timbre (FNMT) española para la firma digital". [3].

1.2.4 Ventajas.

- Son capaces de almacenar información y procesarla.
- Brindan una mayor versatilidad al poder ser programada.
- Pueden ser multiaplicación.
- Reducen el riesgo de fraude.
- En cualquier lugar que las tarjetas Inteligentes reemplacen papel, habrá una reducción de costos.
- Gran capacidad de memoria, con respecto a las anteriores tarjetas (tarjetas de banda magnética).
- Confiabilidad.
- Privacidad.
- Portabilidad.
- Facilidad de usos sin necesidad de conexiones en línea.
- Comodidad para el usuario.
- Cumple con estándares específicos de la ISO 7816.

1.2.5 Desventajas.

- *"Es necesario un lector para las tarjetas inteligentes.*
- *Por su tamaño las tarjetas pueden extraviarse.*
- *Depende de la energía eléctrica para su utilización.*
- *Puede ser dañada si se derrama un líquido sobre ella".* [4]

1.2.6 ¿Por qué son seguras las Tarjetas Inteligentes?

"Las tarjetas inteligentes, en virtud de su poder de cómputo, son sin duda alguna, la tarjeta de tecnología más segura usada en nuestros días. Las tarjetas y las terminales se identifican unas a otras utilizando métodos de autenticidad mutuos. La inserción del chip no puede ser cambiada, y la información que contiene ha sido protegida mediante un control de acceso (PIN)". [4].

- *"Seguridad de los componentes*
  - *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.*
    - *Cambios de frecuencia de reloj.*
- *Seguridad del sistema operativo*
  - *Control de los accesos a memoria*
  - *Protección de datos y ficheros*
- *Seguridad del sistema operativo y de las transacciones.*
  - *Autentificación del portador. (mediante PIN)*
  - *Autentificación de la tarjeta a través de un sistema de claves diversificadas.*
    - *Encriptación/ Desencriptación DES[4].*
    - *Encriptación/ Desencriptación RSA[5].*
    - *Firma digital: MD5[6] ".* [5].

[4] **DES** (Data Encryption Standard o Estándar de Encriptación de Datos). Es el algoritmo de cifrado simétrico más estudiado, mejor conocido y más empleado del mundo.
[5] **RSA** (Rivest, Shamir, Adleman)**.** Es el algoritmo de mayor uso en encriptación asimétrica.
[6] **MD5** (Message Digest 5). Es una función hash irreversible, es decir, codifica la contraseña tecleada por el usuario y es imposible que partiendo de la cadena codificada obtenga la contraseña origen.

1.3 Tipos de Tarjetas Inteligentes.

1.3.1 Tarjetas de Contactos.

Las tarjetas de contacto son las que necesitan ser insertadas en un lector de tarjetas inteligente para que por medio de contactos pueda ser leída. Existen dos tipos de tarjeta inteligente de contacto: Las sincrónicas y las asincrónicas.

Tarjetas de Contactos Sincrónicas o de Memoria.

Estas tarjetas son cargadas previamente con un valor que va decreciendo a medida que se utiliza y una vez que se acaba el monto se vuelve desechable.

Memoria Libre: La información almacena dentro de estas tarjetas no esta protegida por ningún mecanismo de seguridad, lo que las hace utilizable solo en aquellos lugares donde no se necesite una alta seguridad, como: para el pago de peajes, teléfonos públicos, entre otros.

Memoria Protegida: Poseen un circuito de seguridad que proporciona un sistema para controlar los accesos a la memoria frente a usuarios no autorizados. Este sistema funciona mediante el empleo de un código de acceso.

1.3.2 Tarjetas Asincrónicas.

Estas tarjetas poseen en su chip un microprocesador, que además cuenta con algunos elementos adicionales como son: (Ver Figura 1.2):

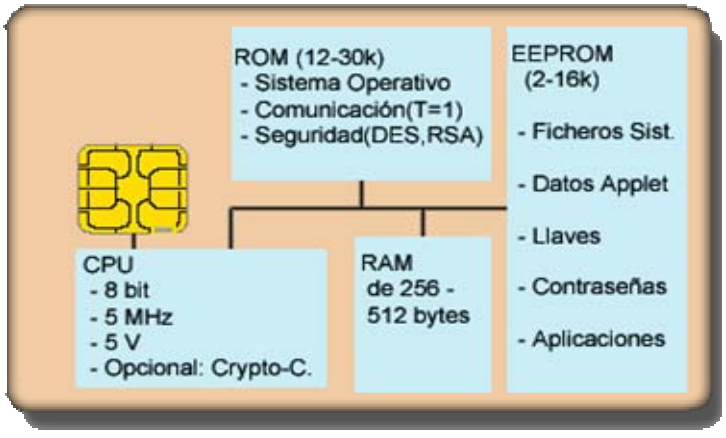


Figura 1.2. (Componentes de una tarjeta inteligente).

- **ROM**: Read Only Memory**.**
- **EEPROM**: Electrical Erasable Programmable Read Only Memory.
- **RAM**: Random Access Memory.
- **I/O:** Puerto de Entrada/Salida.

"El microprocesador y la memoria están fabricados sobre el mismo chip, lo cual hace difícil y caro interceptar las señales que se intercambian entre el procesador y la memoria, proporcionando así una alta seguridad física de los datos almacenados en la memoria". [6]

Memoria ROM.

La memoria ROM es grabada durante la fabricación de la tarjeta y es utilizada para almacenar el Sistema Operativo, datos y aplicaciones de usuarios fijas. Una vez emitida la tarjeta no se puede volver a grabar información en la ROM.

"*Una memoria ROM es aquella que se puede escribir una sola vez y sus datos no se pueden borrar".* [7]

Memoria EEPROM.

Esta memoria es utilizada para gravar información persistente dentro de la tarjeta, ya que preserva el contenido aunque la alimentación se apague. La EEPROM puede ser modifica durante el uso de la tarjeta y los usuarios pueden gravar ahí sus aplicaciones (applets).

"Una EEPROM se borra sometiéndola a radiación ultravioleta. Las tarjetas telefónicas antiguas poseen este tipo de memoria, pero para garantizar la inviolabilidad de la tarjeta, el chip está recubierto de resina opaca que impide el acceso de dicha luz, con lo cual, "en principio", es imposible borrar los datos de la EEPROM. Pero si por algún motivo conseguimos que la luz ultravioleta llegase al chip, lo que haríamos sería borrar por completo los 256 bits de la memoria". [7]

Una gran limitación de la EEPROM es que sufre de desgaste y con la tecnología disponible un bit después de 100,000 escrituras o más deja de ser confiable.

Memoria RAM.

Esta memoria es utilizada para almacenar valores u objetos temporales que los applets necesitan mientras son ejecutados. La RAM es una memoria que no almacena los datos una vez que se quita la alimentación o sea no es persistente.

Puerto de Entrada/Salida.

El puerto de entrada y salida normalmente consiste en un simple registro, a través del cual la información es transferida bit a bit**.**

Contactos de la tarjeta.

Las tarjetas inteligentes poseen ocho contactos en su chip (Figura 1.3), los cuales proporcionan la alimentación y señal de reloj y le permiten recibir y transmitir datos.

Vcc: se utiliza para suministrar la alimentación al chip. El voltaje que se aplica es 3 o 5 voltios, con una desviación máxima del 10 por ciento.

RST: se utiliza para enviar la señal de “reset” al microprocesador.

El microprocesador de la tarjeta inteligente no posee reloj interno. A través del contacto **CLK** se proporciona una señal de reloj externa a partir de la cual se deriva la señal de reloj interno.

CLK: el "reloj" determina la velocidad de funcionamiento de la tarjeta.

GND: es la conexión de masa.

Vpp: se utiliza en tarjetas antiguas para proporcionar el voltaje necesario para programar la EEPROM. En las tarjetas actuales este contacto no se utiliza porque el voltaje se genera internamente.

I/O: se utiliza para transferir datos entre la tarjeta y el dispositivo lector en modo semi-duplex.

RFU: Reservado para su uso en el futuro.

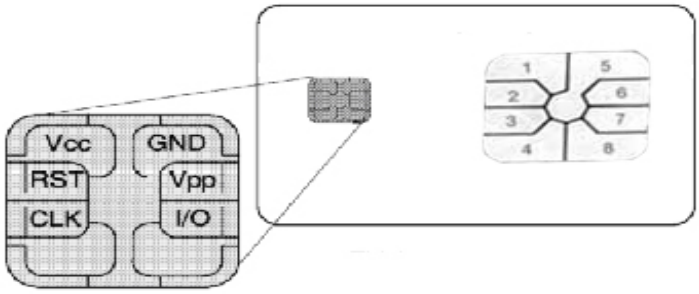


Figura 1.3. (Contactos de una tarjeta inteligente).

1.3.4 Tarjetas Inteligentes sin Contacto.

Estas tarjetas son similares a las de contacto con respecto a lo que pueden hacer y a sus funciones pero utilizan diferentes protocolos de transmisión, el chip se comunica con un lector de tarjetas, puesto

en cada estación, mediante inducción[7] transfiriendo información, a una tasa de transferencia que fluctúa entre 106 y 848 Kb/s, no utilizan contacto galvánico sino de interface inductiva. Poseen además del chip, una antena de la cual se valen para realizar transacciones. Son muy utilizadas en las transacciones que tienen que ser realizadas muy rápidamente.

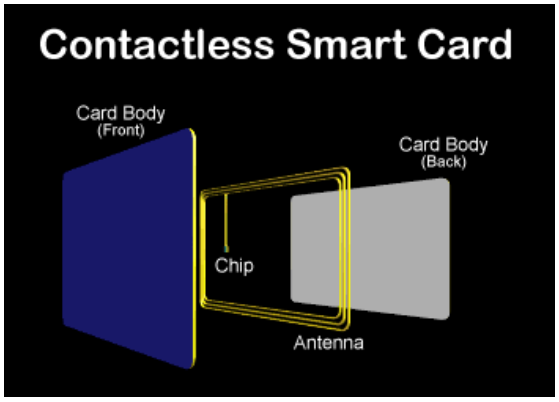


Figura 1.4. (Composición de una tarjeta inteligente sin contacto).

Esta tecnología ofrece ventajas con respecto a la de las tarjetas de contacto. Una de las principales desventajas de las TI con contactos es el deterioro de la superficie de sus contactos. Este problema se resuelve con las TI sin contactos, debido, a que carecen de contactos. Otra de las ventajas es la de no tener que introducir la tarjeta en un lector.

"Este tipo de tarjetas se comunican por medio de radiofrecuencias. Según la proximidad necesaria entre tarjeta y lector, existen dos tipos:

Tarjeta cercana: *Debe estar a unos pocos milímetros del lector para que sea posible la comunicación.*

Tarjeta lejana: *la distancia varía entre centímetros y unos pocos metros".* [8]

Alimentación.

"Desde el punto de vista de cómo se alimentan, existen dos tipos:

- *Uno en el cual la tarjeta incorpora junto al chip una batería que alimenta a los circuitos.*

[7] Fenómeno que origina la producción de una diferencia de potencial eléctrico (o voltaje) en un medio o cuerpo expuesto a un campo magnético variable.

- *Otro tipo que incorpora un hilo metálico incrustado. Este hilo se somete a un campo electromagnético variable que a su vez induce una corriente eléctrica capaz de alimentar los circuitos de la tarjeta".* [8].

1.3.4 Estructura interna de los ficheros de las Tarjetas Inteligentes.

*"Aportado por la norma ISO 7816-4 [8], los S.O. de las tarjetas multi-aplicación soporta los siguientes tipos de archivos, el archivo Maestro (**MF** - Master File) contiene referencias a arreglos de bytes, mientras que un archivo dedicado contiene colecciones de archivos elementales, como un directorio. Además de los dos anteriores, los chipcard de IBM proveen de un Archivo Maestro que corresponde al directorio raíz del sistema de archivos".* [5]

  - **Fichero Maestro (MF):**
    - Representa la raíz de la estructura de ficheros
    - Este archivo es seleccionado automáticamente por el SO del tarjeta después que ocurre un Reset o de un Power Up (momento en que se utiliza el CHIP).
  - **Fichero Dedicado (DF):**
    - Situado debajo del MF
    - Contiene datos relativos a una aplicación
  - **Fichero Elemental (EF):**
    - Situado debajo del MF o de un DF
    - Contiene unidades de datos del sistema o datos de una aplicación
- **Según su estructura interna**
  - **Ficheros transparentes**
    - También llamados binarios
    - Almacenan información no estructurada
    - Los datos se pueden direccionar con un offset
  - **Fichero lineal con registros de tamaño fijo**
    - Los datos son una secuencia de registros individualmente identificables
    - Los registros tienen igual tamaño.
    - Registros direccionados en el orden de su creación
    - No se modifica su número

- **Fichero lineal con registros de tamaño variable**
  - El tamaño de cada registro es fijado mientras se crea
- **Fichero cíclico con registros de tamaño fijo**
  - Los datos son una secuencia de registros individualmente
  - Los registros son direccionados en orden inverso a su creación/modificación.

1.4 Lectores de Tarjetas Inteligentes de Contactos.

Un lector de tarjetas es un dispositivo con una interfaz que permite la comunicación entre una tarjeta y otro dispositivo. Los terminales se diferencian unos de otros en la conexión con el ordenador, la comunicación con la tarjeta y el software que poseen, (Ver Figura 1.5).

1.4.1 Tipos de lectores de Tarjetas Inteligentes:

- **Lectores conectados a un PC:** son lectores fabricados para ser usados conectándolo a un computador, esta conexión puede ser a través de un puerto serie, USB, PCMCIA, etc.
- **Lectores conectados a un equipo específico:** Son lectores que se pueden instalar (previo fabricación y diseño) en un aparato determinado para cumplir con una función. Estos lectores se pueden instalar en:
  - Cajeros automáticos.
  - Máquinas expendedoras.
  - Parquímetros.
  - Puertas (control de acceso).
- **Lectores Portátiles:** Son dispositivos que no necesitan de otro aparato para cumplir su función. Generalmente poseen todos los recursos integrados como baterías, memoria, etc.

1.4.2 Lector escogido.

GemPC USB-SL, (Ver Figura 1.5).

¿Por qué este lector?

- Lee y escribe todas las tarjetas inteligentes con microprocesador de la ISO 7816-1, 2, 3, 4, T=0 y T=1.
- Soporta múltiples plataformas de PC.
- Soporta múltiples lenguajes de programación.

Figura 1.5. (Lector de Tarjetas Inteligentes GemPC USB-SL).

1.5 Protocolos de Comunicación.

1.5.1 APDU

El envío y recepción de datos en una tarjeta inteligente se realiza a través de un formato de paquetes llamados **APDU** (Application Protocol Data Units).

Existen dos tipos de APDU, de comando o de respuesta, con distintos formatos:

- **APDU de Comandos.**

  CLA: identifica que la petición responde al estándar ISO-7816

  INS: código de la instrucción

  P1: primer parámetro

  P2: segundo parámetro

  L_c: número de bytes en el campo de datos

  Data: datos

  L_e: número máximo de bytes esperados en el campo de datos del APDU de respuesta

- **APDU de Respuesta.**

  Data: cuerpo de datos

  SW_1: primer byte de la palabra de estado

  SW_2: primer byte de la palabra de estado

La idea de la palabra de estado, es reportar códigos de retorno en el caso de que se produzcan errores o excepciones.

1.5.2 Comunicación con la Tarjeta.

La comunicación desde y hacia la tarjeta inteligente se lleva cabo por el contacto C7 (figura 1.3), de modo half-duplex. Lo que significa que sólo puede existir comunicación en un sentido, o sea desde la tarjeta o hacia la tarjeta.

"La comunicación siempre es iniciada por la terminal, lo que significa que la interacción entre la tarjeta y la terminal es del tipo cliente-servidor. (Ver Figura 1.6)

Una vez que la tarjeta es insertada en el lector, ésta le proporciona la corriente y el voltaje para funcionar. La tarjeta entonces se reinicia (power-on-reset), y envía un ATR (Answer To Reset) a la terminal. La terminal realiza un análisis sintáctico del ATR, donde obtiene varios parámetros y envía a la tarjeta una instrucción inicial. La tarjeta genera entonces una respuesta y la envía a la terminal. La interacción cliente-servidor continúa de la misma manera hasta que los procesos terminan y la tarjeta es removida de la terminal.

La capa física para el proceso de transmisión está especificado en el estándar ISO / IEC 7916-3. En ese estándar se encuentran definidos los niveles de voltaje para los cuales la señal es interpretada como un 1 o como un 0 lógicos.

Existen varios protocolos diferentes para el intercambio de información entre la tarjeta y la terminal. Estos protocolos están identificados mediante los caracteres "T=" más un número, como se muestra en la tabla siguiente: " [9]

1.5.3 Protocolos.

Protocolo	Descripción
T0	Comunicación asíncrona, half-duplex, orientada a bytes
T1	Comunicación asíncrona, half-duplex, orientada a bloques
T2	Comunicación asíncrona, full-duplex, orientada a bloques
T3	Full-duplex
T4	Comunicación asíncrona, half-duplex, orientada a bytes (expansión del protocolo T = 0)
T5 - T13	Reservados para su uso en el futuro
T14	No es un estándar ISO.
T15	Reservado para su uso en el futuro

Tabla 1.1: Protocolos de comunicación entre tarjetas y terminales.

Los dos protocolos más comunes actualmente son T=0 y T=1, de los cuales el más utilizado es T=0.

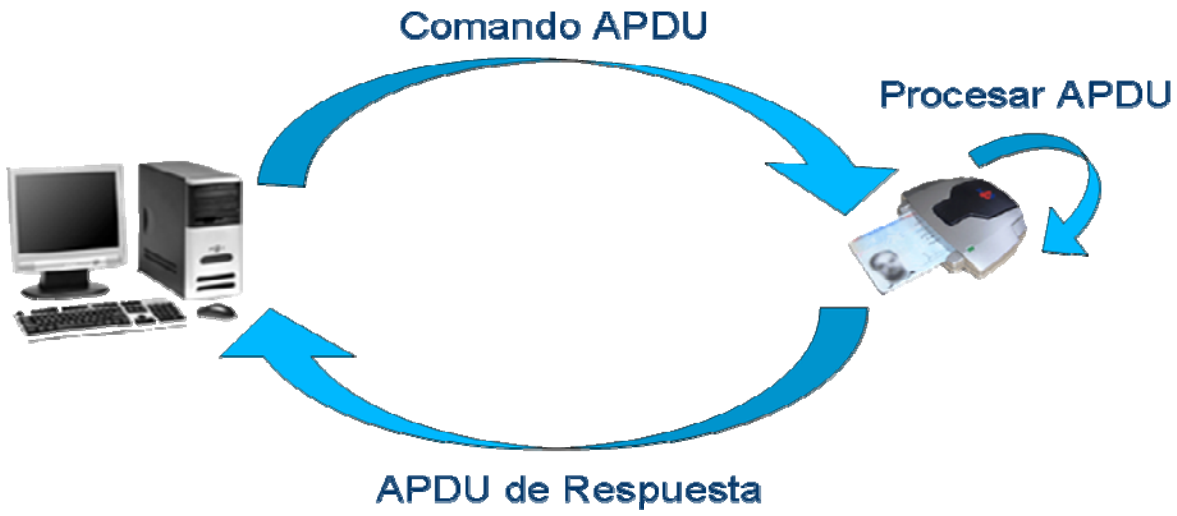


Figura 1.6: Comunicación con la tarjeta.

1.6 Características generales del Sistema Operativo.

"El sistema operativo de una tarjeta inteligente soporta un pequeño conjunto de instrucciones a través del cual se interactúa con la tarjeta. ISO 7816-4 estandariza un amplio rango de instrucciones en forma de APDUs. El sistema operativo de una tarjeta puede soportar todas o algunas de estas APDUs, así como nuevas instrucciones que añada el fabricante". [6]

1.6.1 ¿Por qué Tarjetas inteligentes multiaplicación?

El objetivo principal de las TI multiaplicación es almacenar en una sola tarjeta todas las aplicaciones posibles para así reducir el número de tarjetas en manos del usuario y así eliminar el problema que trae consigo el uso de tarjetas de banda magnéticas. Las tarjetas telefónicas públicas, tarjetas médicas que almacenan las historias clínicas son ejemplos de tarjetas inteligentes de memoria, pero que traen consigo el mismo problema que el uso de las tarjetas magnéticas, el hecho de que el usuario cuente con una tarjeta para cada aplicación.

"…las tarjetas inteligentes tienen la capacidad de integrar todas esas aplicaciones juntas para formar una tarjeta multiaplicación utilizando el procesador y los espacios de almacenamiento de memoria. Sin

embargo este tipo de integración siempre esta limitado más por algunos elementos logísticos que por las capacidades técnicas." [9]

1.6.1 Sistemas Operativos Multi-Aplicación para tarjeta inteligente (MACOS)

- **JavaCard:** Para programación desarrollada en Java.
- **MultOS:** Es el primer COS (Chips Operating System) para tarjeta inteligente que incluye las siguientes características: es abierto, proporciona una alta seguridad y es multiaplicación.
- **Windows para Smart cards:** Sistema operativo de Microsoft para tarjeta inteligente.

1.6.2 Tareas del Sistema Operativo.

"El Sistema Operativo contenido en la tarjeta inteligente se encarga de las siguientes tareas:

- *Transmisión de información a través de la interfaz de comunicación serial.*
- *Carga, operación y administración de aplicaciones.*
- *Procesamiento y control de ejecución de instrucciones.*
- *Administración de la memoria (acceso a la información y manipulación de archivos).*
- *Administración y ejecución de algoritmos criptográficos.*

El tamaño típico de un sistema operativo se encuentra entre los 3 y los 24 Kbytes. Esta variación en el tamaño es debida a que algunas tarjetas son fabricadas para aplicaciones específicas, mientras que otras tienen la capacidad de almacenar diferentes aplicaciones.

Ya que la memoria en este tipo de dispositivos es un recurso muy limitado, no todas las instrucciones y las estructuras de datos se encuentran implementadas en todos los sistemas operativos. Por esta razón, en los estándares ISO 7816-4 y END 726-3 se crearon los llamados "perfiles", que especifican los requerimientos mínimos de estructuras de datos e instrucciones que un sistema operativo para una tarjeta inteligente debe implementar". [9]

1.7 Estándares de la ISO-7816.

En 1987 se publicó el primer estándar para la industria de tarjetas inteligentes, ISO-7816, con el que se intentaba solucionar el problema de interoperabilidad de las tarjetas inteligentes. Por medio de este

estándar se establece la forma y dimensiones de las tarjetas, el significado y localización de los contactos del circuito integrado y el protocolo de comunicación de la tarjeta.

La tarjeta inteligente más básica cumple los estándares de la serie ISO 7816, partes 1 a 10. Este estándar detalla la parte física, eléctrica, mecánica y la interfaz de programación para comunicarse con el microchip.

1.7.1 Descripción de cada una de las partes de la ISO 7816:

- *"7816-1: Características Físicas.*
- *7816-2: Dimensiones y ubicaciones de los contactos*
- *7816-3: Señales Electrónicas y Protocolo de Transmisión*
- *7816-4: Comandos de intercambio inter-industriales*
- *7816-5: Sistema de Numeración y procedimiento de registración*
- *7816-6: Elementos de datos inter-industriales*
- *7816-7: Comandos inter-industriales y Consultas Estructuradas para una Tarjeta*
- *7816-8: Comandos inter-industriales Relacionados con Seguridad.*
- *7816-9: Comandos adicionales inter-industriales y atributos de seguridad.*
- *7816-10: Señales electrónicas y Respuesta al Reset para una Smart Card Síncrona.*
- *Una descripción para las smart cards sin contacto está descrito en el estándar ISO 14443".* [8]

1.8 Utilización de las Tarjetas Inteligentes.

Las tarjetas inteligentes fueron desarrolladas con el objetivo de almacenar información e interactuar con la información contenida en ellas. Estas tarjetas estas dejando atrás a las conocidas tarjetas de banda magnética, debido a su capacidad de poder modificar el contenido y de realizar múltiples grabaciones, sin temor a perder la información contenida en ellas.

Debido a las ventajas que estas ofrecen, se ha ido observando a nivel mundial un incremento de la utilización de las TI desde su aparición. Hoy en día suelen utilizarse para implementar módulos de seguridad y entre sus principales aplicaciones se encuentran los sectores bancarios, de telefonía móvil y de comercio electrónico, aunque son muy utilizadas en aplicaciones de la salud.

El uso de la TI ya está probado, algunos países como Francia, Bélgica y los Estados Unidos, están utilizando esta tecnologías, para el almacenamiento de la información de salud y como identificación personal.

"En Bélgica más de un millón personas disponen de carné de identidad electrónicos basados en la tecnología JavaCard, actualmente se están emitiendo unos 150.000 carné al mes y se espera que para finales de 2009, 8,2 millones de ciudadanos mayores de 12 años tengan su carne de identidad". [10].

1.9 Aplicaciones.

La realización de software asociado a este nuevo entorno permite diversidad de aplicaciones comerciales. El primer gran mercado que utilizó las tarjetas con chip fueron las tarjetas telefónicas y las tarjetas bancarias.

Algunos tipos de aplicaciones con tarjetas inteligentes son:

- Monederos Electrónicos.
- Control de Seguridad de Acceso.
- Tarjetas Telefónicas.
- Tarjetas de Salud.
- Tarjetas de Seguro Social.
- Pagos seguros por Internet.
- Tarjetas de Crédito/Debito.
- Transporte.

1.10 JavaCard.

Dentro de la categoría de tarjeta inteligente con microprocesador se encuentran las llamadas JavaCard o Java Smart Card. El tamaño y el protocolo de comunicación de las tarjetas inteligentes están estandarizados, pero son los fabricantes los que determinan su funcionamiento interno. Los programadores de tarjetas inteligentes tienen que trabajar con protocolos de comunicación de bajo nivel, gestión de memoria y detalles de hardware dependientes de la implementación. Una JavaCard es una tarjeta inteligente capaz de ejecutar programas desarrollados en Java. *"La primera JavaCard en salir al mercado fue producida por Schlumberger*[8]*, aún antes de que Sun fijara el estándar".* [1]

[8] Grupo industrial internacional que ofrece productos y servicios. Desarrolla y fabrica las tarjetas (con y sin chips) y los terminales asociados que permiten optimizar la seguridad en las transacciones electrónicas.

Esta tecnología combina parte del lenguaje de programación Java con un entorno de ejecución optimizado para Tarjeta Inteligente y similares. El objetivo de la tecnología JavaCard es llevar los beneficios del desarrollo de software en Java al mundo de las Tarjeta Inteligente y permitir el desarrollo de aplicaciones en estas. Las principales complicaciones tienen que ver con lo limitado de los recursos de hardware.

"El primer intento de aplicar Java a las tarjetas inteligentes se produjo en 1996, cuando un grupo de ingenieros de Schlumberger propusieron el primer API JavaCard. Pocos meses después, varios fabricantes de tarjetas, entre ellos Bull y Gemplus, se unieron a Schlumberger en el JavaCard Forum, con el objetivo de desarrollar y promover la tecnología JavaCard. A finales de 1997, Sun Microsystems anunció la especificación JavaCard 2.0 ".[6]

"Los componentes principales dentro de una JavaCard son el microprocesador y las memorias. La arquitectura básica de una JavaCard consiste de Applets, JavaCard API, JavaCard Virtual Machine (JCVM) JavaCard Runtime Environment (JCRE) y el sistema operativo nativo de la tarjeta". [1] . (Ver Figura 1.6)

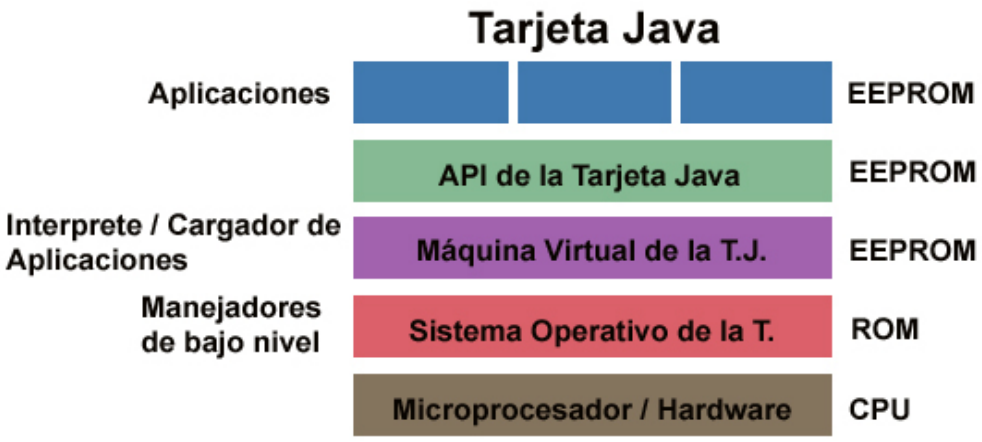


Figura 1.7: Estructura de una Tarjeta Java

1.10.1 Componentes de una Java Card.

- **Java Card Virtual Machine (JCVM).**

Define la máquina virtual y el lenguaje de programación Java adecuado a las tarjetas inteligentes. JCVM fue basado en la maquina virtual de java.

- **Java Card Runtime Environment (JCRE).**

Describe el comportamiento en ejecución de JavaCard, incluyendo la gestión de memoria y de las aplicaciones (applets).

- **Java Card Application Programming Interface (API).**

Describe los paquetes y clases Java para programar aplicaciones en tarjetas inteligentes.

1.10.2 Lenguaje JavaCard.

La plataforma JavaCard soporta un limitado subconjunto de los elementos del lenguaje de programación Java esto se debe a la limitación de memoria y de capacidad de procesamiento de las tarjetas inteligentes, sin embargo, se puede utilizar un compilador de Java para compilar ficheros fuente JavaCard, cambiando las clases estándar del Kit de desarrollo de Java (JDK) por las del entorno JavaCard. (Tabla 1.3).

Paquetes	Descripción
java.io	Subconjunto del paquete *java.io* de Java estándar.
java.lang	Subconjunto del paquete *java.lang* de Java estándar.
javacard.framework	Funcionalidad básica para applets.
javacard.framework.service	Servicio de componentes.
javacard.security	Funciones criptográfica.

Tabla 1.3: Paquetes JavaCard.

1.10.3 Máquina Virtual de JavaCard.

La Máquina Virtual de JavaCard (JCVM) es una adaptación de la Máquina Virtual de Java para tarjetas inteligentes, debido a que los recursos que disponen las tarjetas son escasos, la JCVM se ha dividido en dos partes: una que se ejecuta en la tarjeta, el intérprete, y otra que se ejecuta fuera de ella, el conversor.

La JCVM esta activa en la tarjeta mientras la tarjeta este conectada al lector, una vez quitada la alimentación, la JCMV deja de funcionar temporalmente hasta que no es conectada nuevamente, donde recupera toda la información almacenada en la tarjeta.

El conversor es el encargado de realizar fuera de la tarjeta, donde no hay limitaciones de recursos, las tareas de la Máquina Virtual que no necesitan realizarse en tiempo de ejecución, como la carga de clases y la verificación y optimización del código de bytes. Básicamente, el conversor carga y pre-procesa los ficheros class de un paquete Java y los encapsula en un fichero de formato especial, un fichero CAP[9], para que sea descargado a la tarjeta y ejecutado por el intérprete, que es la parte de la Máquina Virtual interna a la tarjeta.(Ver Figura 1.8)

La conversión elimina una cantidad significativa de datos en tiempo de ejecución, relacionado con la carga dinámica de las clases. El propósito principal de la TI es evitar las conexiones on-line, por lo que la plataforma JavaCard no necesita soportar esta carga. Todas la clases de un paquete JavaCard se descargan al mismo tiempo en la TI.

Las clases de un paquete JavaCard son convertidas a un fichero CAP. Este fichero contiene las clases del paquete JavaCard y esta optimizado para obtener un mejor rendimiento y minimizar el tamaño de lo ficheros en la tarjeta. Para lo que se realizan los siguientes procedimientos de optimización:

- La tecnología JavaCard solo soporta datos de tipo short, lo que conlleva a un mejor aprovechamiento de la memoria y a un mejor rendimiento de esta tecnología.
- Para minimizar el tiempo de descarga de las aplicaciones a la tarjeta, es necesario minimizar el tamaño de los ficheros. La referencia simbólica de las clases, métodos, y variables son convertidas a identificadores de tipo short, para que la tarjeta los maneje de una forma más eficiente.

La Máquina Virtual de JavaCard que reside en la tarjeta, el intérprete, proporciona el soporte del lenguaje Java en la tarjeta, de forma que los applets puedan ejecutarse independientemente del hardware de la tarjeta. El intérprete realiza tres tareas fundamentales:

[9] CAP: Converted Applet (applet convertido).

- Ejecuta las instrucciones indicadas por el código de bytes (fichero CAP).
- Controla la asignación de memoria y la creación de objetos.
- Toma parte en las tareas de seguridad.

El intérprete ejecuta el código que encuentra en los ficheros CAP, pero es el instalador de la plataforma JavaCard el encargado de descargar e instalar los ficheros CAP en la tarjeta. El instalador reside en la tarjeta y coopera con un programa de instalación fuera de la tarjeta que, a través del dispositivo lector, le transmite el código binario ejecutable de un fichero CAP. El instalador escribe el código en la memoria EEPROM, realiza el enlace con otras clases que residan ya en la tarjeta, y crea e inicializa las estructuras de datos que el Entorno de Ejecución de JavaCard (JCRE) utiliza internamente.

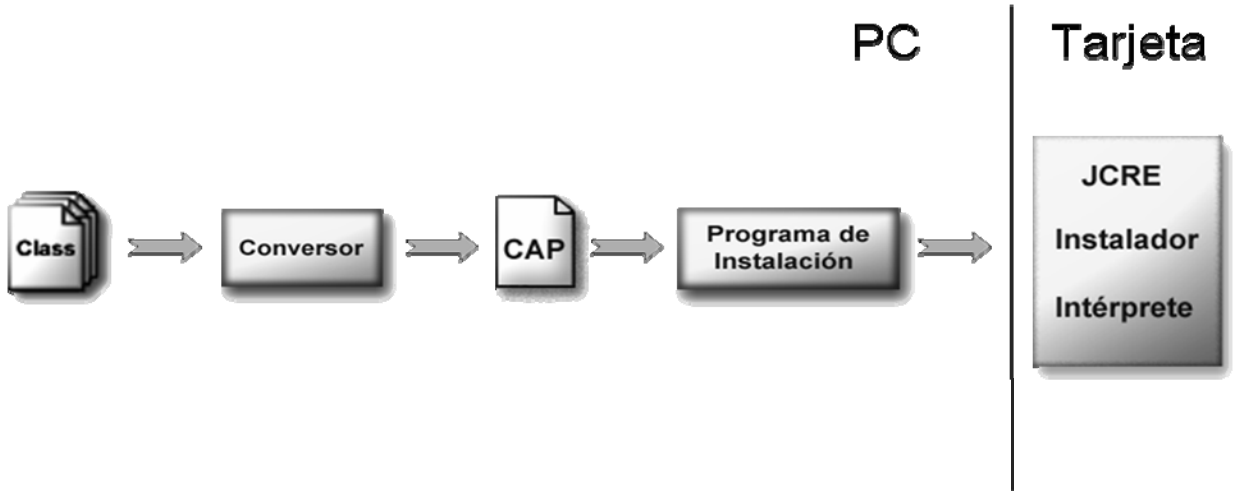


Figura 1.8: Instalación de un applet.

1.10.4 JavaCard Runtime Environment (JCRE).

La clase javacard.framework.Applet define cuatro métodos públicos que son utilizados por el JCRE para hacer funcionar las aplicaciones.

- **Método install(byte[], short, byte)**

Cuando el método Install() es llamado , el applet no ha sido instanciado todavía. Este método llama al constructor de la subclase Applet para registrar una instancia del applet, crear los objetos que el applet

necesita para su ejecución, y por ultimo instalar el applet con el método install(). Típicamente, un applet crea varios objetos, los inicializa con valores predefinidos, cambia el estado de algunas variables internas y llama al método Applet.register() o al Applet.register(byte[], short, byte) para especificar el AID (applet IDentifier como se definió en la ISO 7816-5) para ser usados para su selección. Esta instalación es considerada exitosa cuando la llamada al Applet.register() se completa sin ninguna excepción. La instalación es considerada no exitosa si el método install() no llama al método Applet.register(), o si se lanza una excepción en el método install().Si la instalación no es exitosa, la JCRE realizar una limpieza cuando recobra el control. Esto significa que todas las actualizaciones condicionales del almacenamiento persistente deben ser retornadas al estado que tenían previo a la llamada del método install(). Si la instalación fue exitosa, la JCRE marca al applet como disponible para seleccionar.

- **Método select()**

En un tarjeta pueden coexistir varios applets estos se encuentran en un estado suspendido hasta que son seleccionados. La selección ocurre cuando la JCRE recibe un comando SELECT FILE APDU en el cual se especifica el AID del applet que se desea seleccionar. Si al seleccionar un applet se encuentra otro applet seleccionado, el JCRE es el encargado de deseleccionar dicho applet con el método deselect(), y luego invoca al método select() del applet cuyo AID fue especificado. Un applet puede rechazar la selección en cuyo caso el JCRE es el responsable de responder adecuadamente al CAD (Card Acceptance Device). En caso de fallar la selección, el estado del JCRE es cambiado para indicar que ningún applet fue seleccionado. Si el método select() retorna true entonces se llama al método process(), del applet seleccionado para que lo procese y devuelva al CAD la información que sea pertinente.

- **Método process(APDU)**

Todas las APDUs son recibidas por JCRE y pre-procesadas, este invoca al método process(APDU) del applet seleccionado pasándole como parámetro el COMMAND APDU recibido.

En caso de que la ejecución finalice correctamente, el applet sólo debe encargarse de cargar en el RESPONSE APDU la información que va a devolver, si la hay. El JCRE es responsable de setear los SW del RESPONSE APDU al valor especificado para ejecución exitosa (0x9000, de acuerdo a lo especificado en el ISO 7816). Durante el proceso de un APDU, el applet puede levantar una ISOException con los SW apropiados.

- **Método deselect()**

Cuando el JCRE recibe un comando SELECT FILE APDU donde se le especifica el AID de un applet, (aún cuando el AID del applet a seleccionar coincida con el del applet seleccionado), el JCRE llama al método

deselect() del applet que se encuentra seleccionado. El método deselect() permite al applet llevar a cabo las operaciones de limpieza que puedan ser requeridas para permitirle a otros applet su ejecución.

- **Método uninstall()**

Este método es definido en la interfaz javacard.framework.AppletEvent. Cuando el JCRE se esta preparando para borrar un instancia de un applet, el JCRE llama al método uninstall(), si es implementado por el applet, para informarle a este de la petición de borrado. En el retorno de este método, el JCRE chequea por referencias de dependencias antes de borrar la instancia del applet. Este método es llamado múltiples veces para cada intento de borrado de un applet.

1.10.5 Entorno de Ejecución JavaCard

El JCRE es el equivalente al Sistema Operativo de la tarjeta, ya que se ejecuta en la tarjeta y es el responsable de la gestión de recursos de esta, la ejecución de applets y la seguridad entre los applets.
El Entorno de Ejecución de JavaCard (JCRE) se sostiene sobre los siguientes componentes (Figura 1.6):

- La Máquina Virtual de JavaCard (el intérprete)
- Las clases API de JavaCard
- Extensiones específicas del fabricante
- Las clases de sistema

Los métodos nativos dan soporte a la JCVM y a las clases del sistema, y son los responsables de manejar, los protocolos de comunicación y la asignación de memoria, la criptografía, etc. Estas clases son las encargadas de gestionar las transacciones, la comunicación entre los applet y las aplicaciones fuera de la tarjeta, así como controlar la creación y selección de los applets.

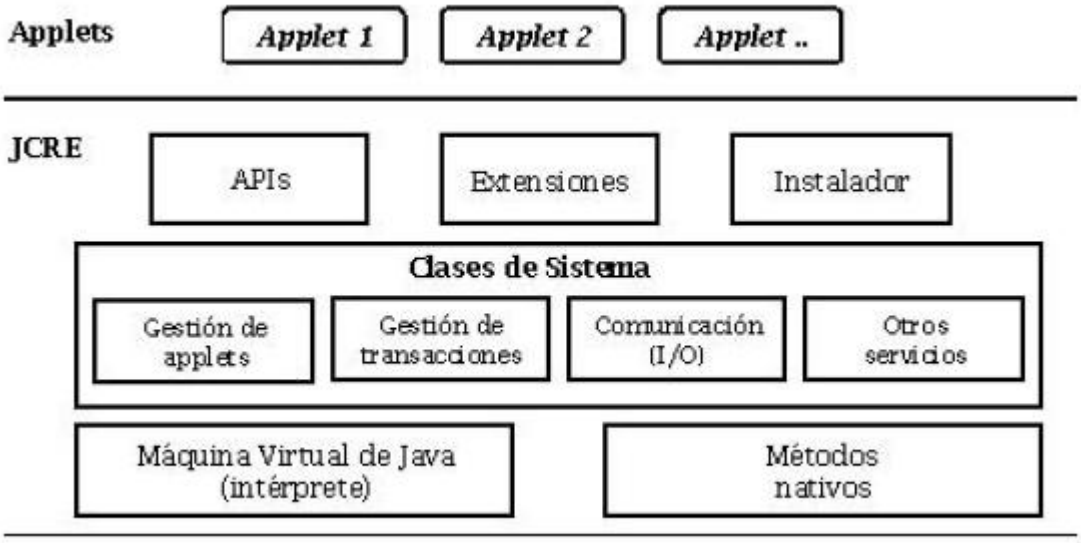


Figura 1.9: (Hardware y Sistema Nativo de una Tarjeta Inteligente).

Después que la tarjeta es fabricada hay que instalarle los applets, de los servicios que esta brindará. De esta función se encarga el instalador, el cual de forma segura descarga los applets a la tarjeta.

Una vez que la tarjeta es inserta en el lector, este le proporciona la alimentación al reloj y al reset, luego el JCRE se mantiene a la espera de recibir comandos APDU, cuando se recibe un comando el JCRE es el encargado de seleccionar un applet, si es que en el comando APDU viene el AID de un applet, o envía el comando a un applet si ya existe uno seleccionado con anterioridad, para que lo procese previamente. Luego el applet envía la respuesta a través de otra APDU y sede el control al JCRE, el cual se mantiene a la espera de nuevos comandos.

Entre las características destacadas del JCRE hay que mencionar las siguientes:

- **Objetos persistentes y transitorios.** Los applet pueden crear objetos en la RAM y la EEPROM, los objetos creados en la RAM se usan para la seguridad o para tener un mejor rendimiento, estos son destruidos una vez quitada la alimentación. El propósito de este requisito es permitir que los objetos transitorios sean utilizados para almacenar llaves de la sesión.
- **Operaciones y transacciones atómicas.** Definen como la tarjeta maneja la información persistente después de una parada. Una transacción es un sistema lógico de las actualizaciones de datos persistentes, es importante que las transacciones sean atómico: lo que significa que todas las zonas de información son actualizadas, o ningunas lo son. Si la transacción no termina

correctamente el JCRE proporciona ayuda para restaurar los datos de la tarjeta a su estado original de la pre-transacción.

Firewall. Una tarjeta java puede contener varios applets, cada applet tiene su propio espacio de memoria, o sea, están protegidos por un Firewall lo que significa que un applet no puede tener acceso a los campos o a los objetos de un applet en otro contexto.

Un applet puede obtener información de otro applet, pero el applet de petición debe satisfacer ciertas reglas antes de poder acceder a la información, estas reglas son proporcionadas por el JCRE a través del API de JavaCard, el cual contiene mecanismos bien definidos para accedes de forma segura a los métodos de otros applets.

El Firewall también proporciona seguridad ante el código incorrecto, para que los applet no puedan ser alcanzados por dicho código. (Ver Figura 1.7).

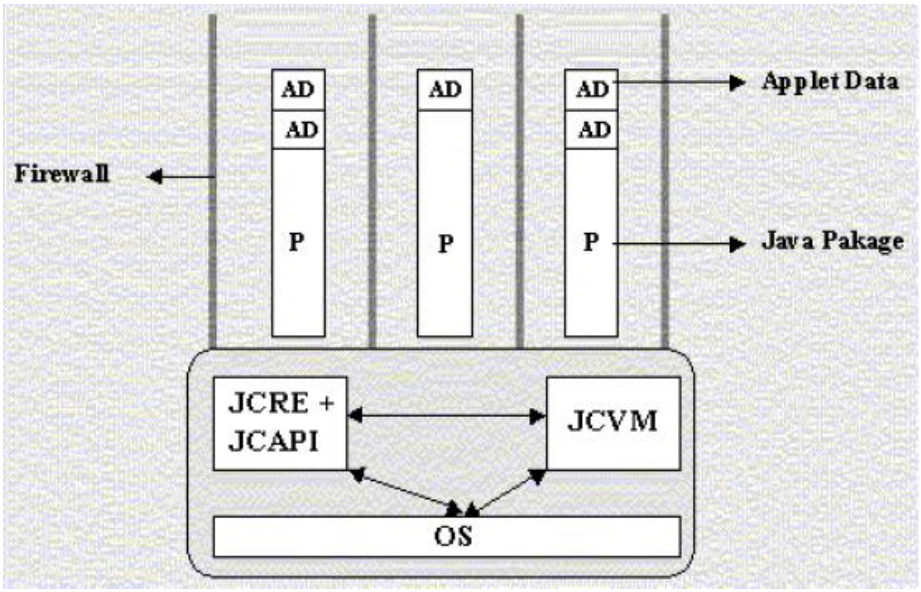


Figura 1.10: Firewall de una Tarjeta Inteligente Java.

1.10.6 Seguridad.

Seguridad física.

Actualmente muchas personas prefieren a las Tarjetas Inteligentes, ya que estas son muy seguras y difíciles de atacar, física o lógicamente, ya que estas disponen de mecanismos de protección ante diferentes ataques como son: detección de ciclos de reloj anormales en frecuencia, retiro de la cubierta de resina epoxi o exposición del microprocesador a luz ultra violeta.

Sin embargo, en ocasiones se ha logrado obtener información a través de diversos medios, estos ataque sucedieron hace mucho tiempo y los fabricantes asegura que sus componente son cada vez son mas seguros lo que deja sin posibilidades de obtener información a los atacantes.

Seguridad lógica.

Actualmente los usuarios finales poseen gran cantidad de tarjetas (una para cada aplicación), estos desean reducir este número, y las Tarjetas Inteligentes Multiaplicación le proporcionan esta ventaja.

Las Tarjetas Inteligentes brindan la posibilidad de cooperación entre diferentes empresas o ministerios, para albergar en una misma tarjeta diferentes servicios al usuario final.

1.11 Principales Fabricantes.

Gemplus (www.gemplus.com)

Schlumberger (www.slb.com)

Bull (www.bull.com)

Oberthur (www.oberthur.com)

Orga (www.orga.com)

Solaic (www.winforms.phil.tu-bs.de/winforms/company/solaic/solaic.html)

De la Rue (www.delarue.com)

1.12 Emulador de EclipseJCDE.

Eclipse es una comunidad de código abierto cuyos proyectos se centran en proporcionar una plataforma de desarrollo independiente de los fabricantes así como aplicaciones que sirvan de marco de trabajo para el desarrollo de software.

Eclipse proporciona un marco de trabajo basado en plug-ins que facilita la creación, integración y utilización de herramientas software.

La plataforma Eclipse está escrita en lenguaje Java y viene con una gran cantidad de herramientas y ejemplos para la construcción de plug-ins.

EclipseJCDE es un conjunto de plug-ins para eclipse que envuelve el Kit de Desarrollo de JavaCard (JCDK) de Sun Microsystems para proporcionar un entorno visual en el que desarrollar aplicaciones Java Card, automatizando muchos de los pasos necesarios.

Utilizando el emulador JCWDE (Java Card Workstation Development Environment), que es un emulador de applets, disponible en el Kit de JCDK, es que el EclipseJCDE visualiza el comportamiento del los applets que se desarrollan en su entorno.

El único requisito básico para el funcionamiento de Eclipse es tener instalado un entorno de ejecución de Java, JavaTM 2 SDK, Standard Edition.

1.13 Conclusiones.

En el transcurso de este capitulo se ha hecho un estudio de la tecnología de las tarjetas inteligentes, especialmente de las JavaCard, se ha dado a conocer las diferentes tarjetas que existen, también se ha descrito el funcionamiento de las mismas y los protocolos que usan para su comunicación, para así lograr un mejor entendimiento de las mismas.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción.

En este capítulo se describen la realización de los Casos de Usos, se definen los actores y trabajadores que participan en el negocio, quedan explícitas las reglas del negocio, y se brinda además una representación gráfica del Modelo de negocio. Además se realiza la descripción del sistema a automatizar, se hace un diagrama de casos de uso del sistema para un mejor entendimiento de como se va a informatizar el mismo. Se definen los actores del sistema y los requerimientos que debe cumplir la aplicación de software que se implementó. Se definen los requerimientos mínimos que debe tener el sistema de cómputo donde se vaya a instalar la aplicación.

2.2 Información que se maneja.

En cuanto a la información del paciente que se maneja se encuentra los datos de identificación del paciente, las enfermedades, las alergias, así como datos relacionados a las consultas. Ver [Anexo 1].

2.3 Reglas del negocio.

Reglas de negocio a considerar:

- El médico tendrá que haberse autenticado antes de realizar cualquier operación.
- El paciente deberá tener la Historia Clínica para ser atendido.
- El médico creara una Historia Clínica a los pacientes que no la tengan.

2.4 Actores del negocio.

Nombre del actor	Descripción
Paciente	El paciente es el que asiste a la consulta, y entregar los datos que se le pidan.
Persona	La persona es la encargada de asistir a la oficina de personalización de la tarjeta y entregar los datos que se le pidan.

Tabla 2.1. Descripción de los actores del negocio.

2.5 Diagrama de casos de uso del negocio.

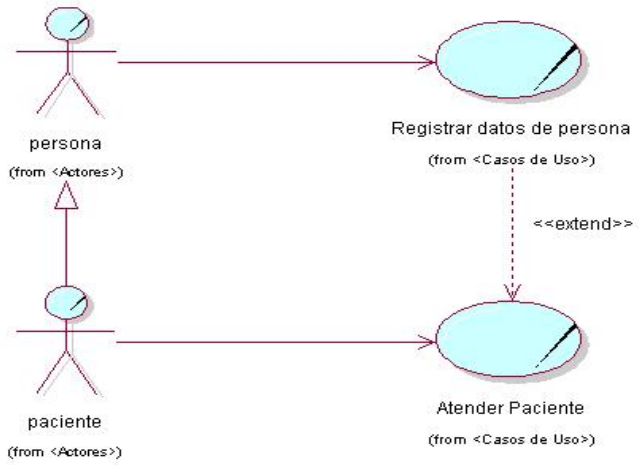


Figura 2.1: Diagrama de Casos de Uso del Negocio.

2.6 Trabajadores del negocio.

Nombre del trabajador	Descripción
Médico	Es el encargado de recibir al paciente en el consultorio, preguntarle sus datos personales, dolencias, confeccionar la HC y realizarle el diagnóstico. No se beneficia en ningún momento de las acciones realizadas en los procesos de negocio que tienen lugar en la clínica, sino que se limita a ejecutar dichas acciones.
SistemaSalud	Es el encargado de almacenar los datos referentes a las personas y a la HC.
Registrador	Es el encargado de personalizar la Tarjeta.

Tabla 2.2. Descripción de los trabajadores del negocio.

2.7 Descripción de los Casos de uso del Negocio.

- Descripción del Caso de uso del Negocio Registrar Datos de la Persona. [Anexo 2].
- Diagrama Actividad del Caso de Uso del Negocio Registrar Datos de la Persona. [Anexo 3].
- Descripción del Caso de uso del Negocio Atender Paciente. [Anexo 4].
- Diagrama Actividad del Caso de Uso del Negocio Atender Paciente. [Anexo 5].

2.8 Modelo de objetos.

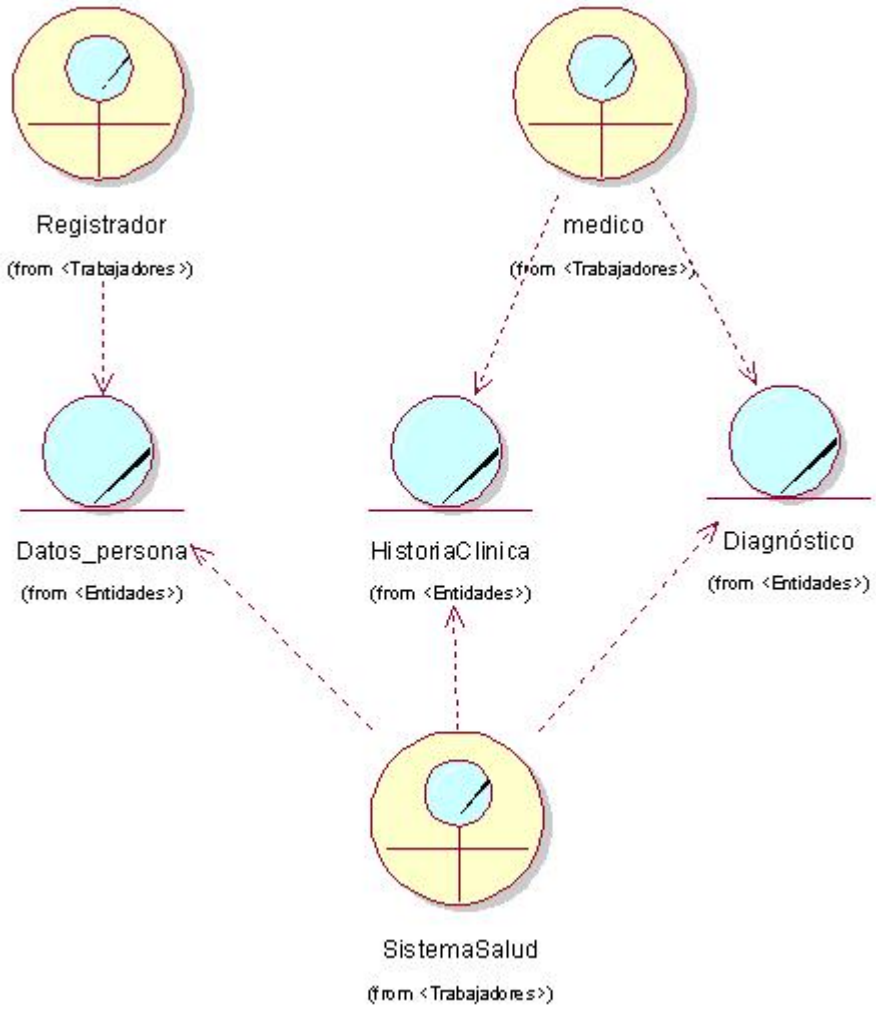


Figura 2.2: (Modelo de Objetos).

2.9 Requerimientos.

2.9.1 Propuesta del sistema.

1. El médico se autentifica.
2. Se permite que el médico cree una HC en caso que no exista.
3. Permite mostrar y actualizar los datos de la HC localizada en la tarjeta.

2.9.2 Trabajadores del Sistema.

Nombre del actor	Descripción
Médico	El Médico es el encargado de interactuar con el sistema para gestionar la información de los pacientes en la HC.
SistemaSalud	Es el encargado de almacenar los datos referentes a las personas y a la HC.
Registrador	Es el encargado de personalizar la Tarjeta.

Tabla 2.3: Definición de actores del sistema.

2.9.3 Requisitos Funcionales.

Los requisitos funcionales del sistema son:

- R1- Registrar datos de la persona.
- R2- Crear HC.
- R3- Realizar Consulta.
- R4- Buscar HC.
- R5- Autenticarse.
- R6- Obtener HC.

2.9.4 Diagrama de Casos de Uso del Sistema.

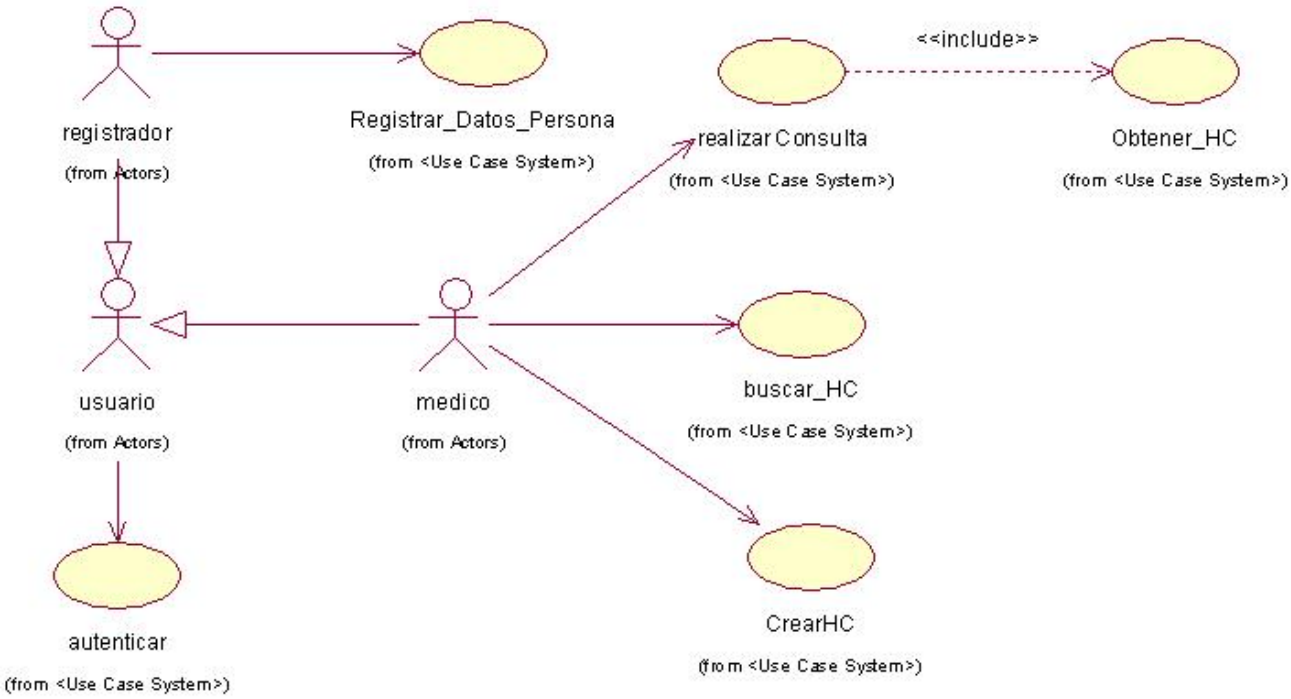


Figura 2.3: (Diagrama de Casos de Uso del Sistema).

- Descripción del Caso de uso del Sistema Autenticar. [Anexo 6].
- Descripción del Caso de uso del Sistema Registrar Datos Persona. [Anexo 7].
- Descripción del Caso de uso del Sistema Crear HC. [Anexo 8].
- Descripción del Caso de Uso del Sistema Realizar Consulta. [Anexo 9].
- Descripción del Caso de uso del Sistema Buscar HC. [Anexo 10].
- Descripción del Caso de Uso del Sistema Obtener HC. [Anexo 11].

2.9.5 Requisitos no funcionales.

Apariencia o interfaz externa.

- La aplicación deberá tener una interfaz externa amigable, que sea sencilla y fácil de entender por el usuario para así evitar que el usuario se pierda dentro de la aplicación.

Software:

- No hay restricciones en cuanto al sistema operativo a instalar puesto que la aplicación será multiplataforma.
- La Máquina Virtual de Java tiene que estar instalada.
- Los Drivers para los lectores de las tarjetas tienen que estar instalados.
- La Máquina Virtual de JavaCard que se ejecuta en la tarjeta debe coincidir con una versión igual o superior a la JCDK

Hardware:

- Se debe contar con 256 MB de memoria RAM como mínimo, aunque lo ideal serian 512 MB.
- Procesadores Pentium IV.
- Lector de Tarjetas Inteligentes.
- Tarjeta Inteligente Java.

Requerimientos en el diseño de implementación:

- Se utilizó el lenguaje de programación JavaCard.
- Se utilizó Eclipse como herramienta de desarrollo.
- Se utilizó Racional Rose para el análisis y diseño de la aplicación.
- Se utilizo el Kit de desarrollo de JavaCard.

Seguridad:

- Confidencialidad: Se requiere de un PIN para poder acceder a la información dentro de la Tarjeta y que el médico este autentificado. El médico es el único que puede modificar la información dentro de la Historia Clínica.
- Disponibilidad: La información siempre estará disponible ya que no se necesitan de conexiones online para poder acceder a la información.

Portabilidad:

- El sistema deberá funcionar en los sistemas operativos Windows y Linux, pero para ello se deberá tener instalada la máquina virtual de Java y los Drivers para los lectores de las tarjetas.

2.10 Conclusiones:

En el transcurso de este capítulo se ha descrito el proceso de creación de una Historia Clínica; donde se identificaron los roles, entidades u objetos del negocio, así como su relación con este proceso. Esta descripción fue realizada mediante el modelado del negocio, para lo cual se elaboraron los modelos de casos de uso y objetos del negocio.

Después de haber realizado el modelado de negocio, se pudo lograr una mejor comprensión del problema que el sistema tiene que resolver.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción.

Tras la definición y descripción, en el capítulo anterior, de las funcionalidades deseadas y necesarias del sistema propuesto; se hace necesario definir cómo se desarrolla. En este capítulo se realiza el diagrama de clases del diseño dando respuesta a la solución que se propone, se da seguimiento y realización a los casos de uso del sistema. Se describen además los principios de diseño que se tendrán durante el desarrollo de la aplicación.

3.2 Diagramas de Clases del Análisis.

Diagrama de clases del análisis.

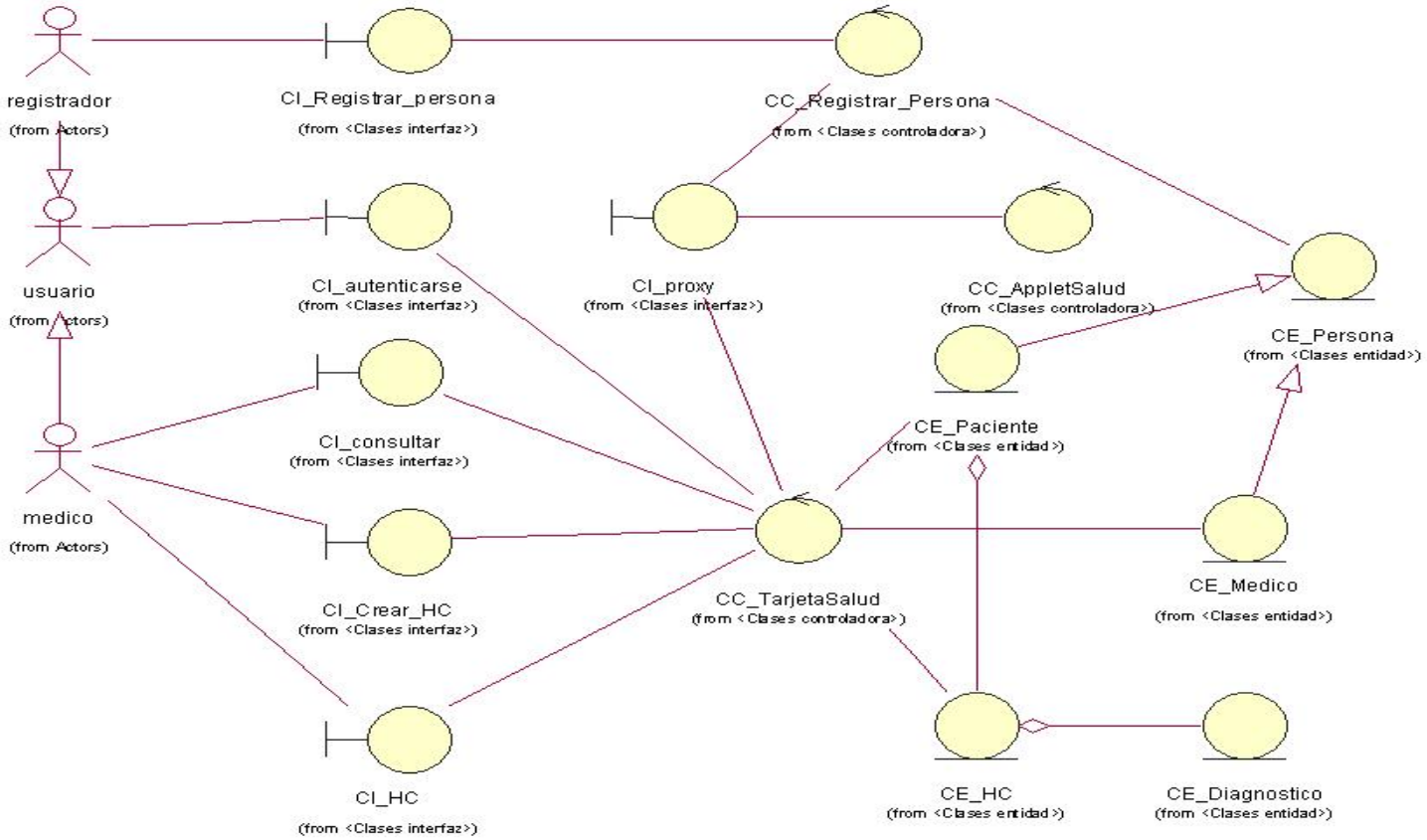


Figura 3.1: Diagrama de clases del análisis

- Descripción de los casos de uso del sistema. [Anexo 12].
- Diagrama colaboración del análisis Autenticarse. [Anexo 13].
- Diagrama colaboración del análisis Autenticarse. Fallido. [Anexo 14].
- Diagrama colaboración del análisis Buscar HC. [Anexo 15].
- Diagrama colaboración del análisis Realizar Consulta. [Anexo 16].
- Diagrama colaboración del análisis Crear HC. [Anexo 17].
- Diagrama colaboración del análisis Registrar Datos Persona. [Anexo 18].
- Diagrama colaboración del análisis Obtener HC. [Anexo 19].

3.3 Diagrama de Clases del Diseño.

Diagrama de Clases del Diseño.

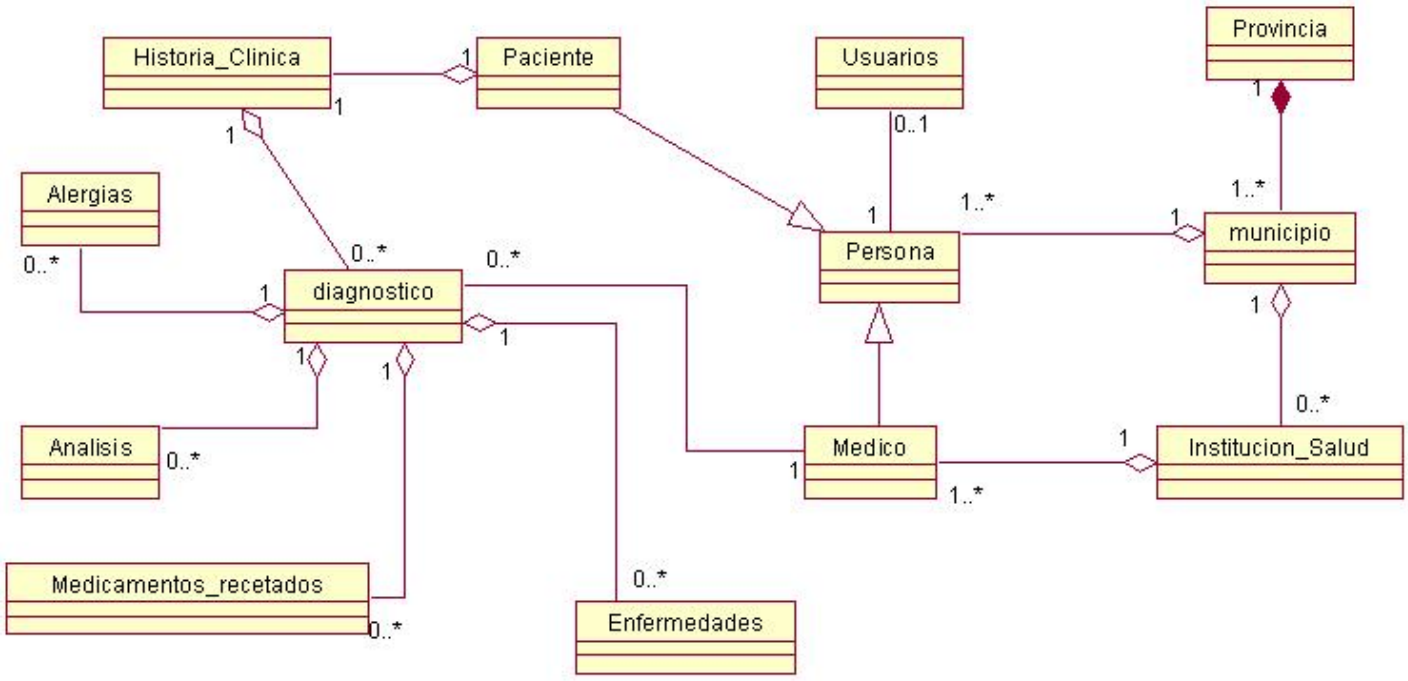


Figura 3.2: Diagrama de clases del diseño.

- Diagrama colaboración del diseño Autenticarse. [Anexo 20].
- Diagrama colaboración del diseño Buscar HC. [Anexo 21].
- Diagrama colaboración del diseño Crear HC. [Anexo 22].

- Diagrama colaboración del diseño Realizar Consulta. [Anexo 23].
- Diagrama colaboración del diseño Obtener HC. [Anexo 24].
- Diagrama colaboración del diseño Registrar Datos Persona. [Anexo 25].

3.3.1 Descripción de las clases del Diseño.

Nombre:	Alergias	
Tipo de clase	entidad	
Atributo		**Tipo**
Id_alergia		Char
Nombre_alergia		Char
Para cada responsabilidad:		
Nombre:	Get_id_alergia()	
Descripción:	Obtiene el id de la alergia	
Nombre:	Get_nombre_alergia()	
Descripción:	Obtiene el nombre de la alergia	

- Para ver las demás descripciones consultar el [Anexo 26].

3.4 Diagrama de Clases de la Tarjeta Inteligente.

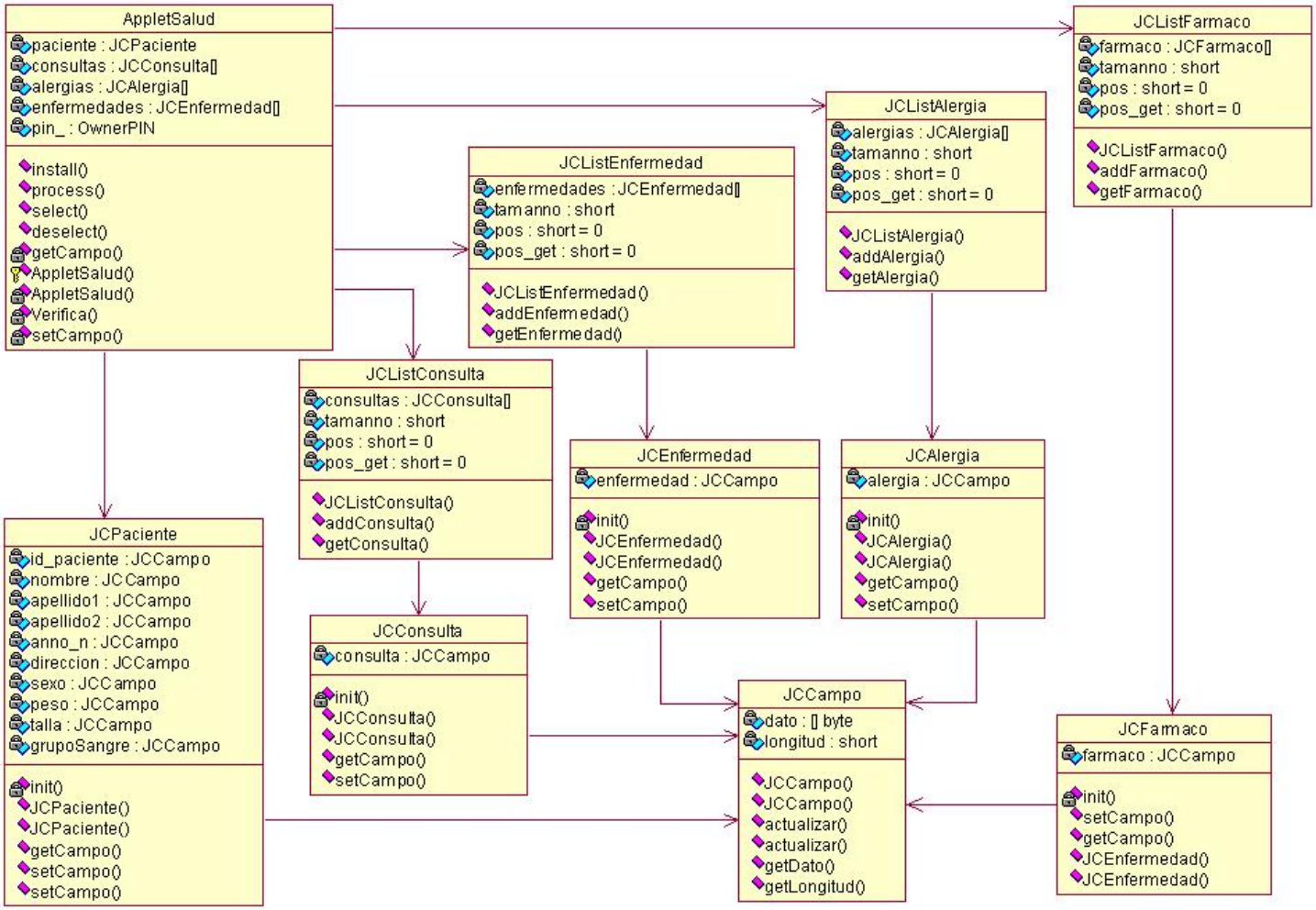


Figura 3.3: Diagrama de Clases de la Tarjeta Inteligente.

3.5 Modelo lógico de la Base de Datos.

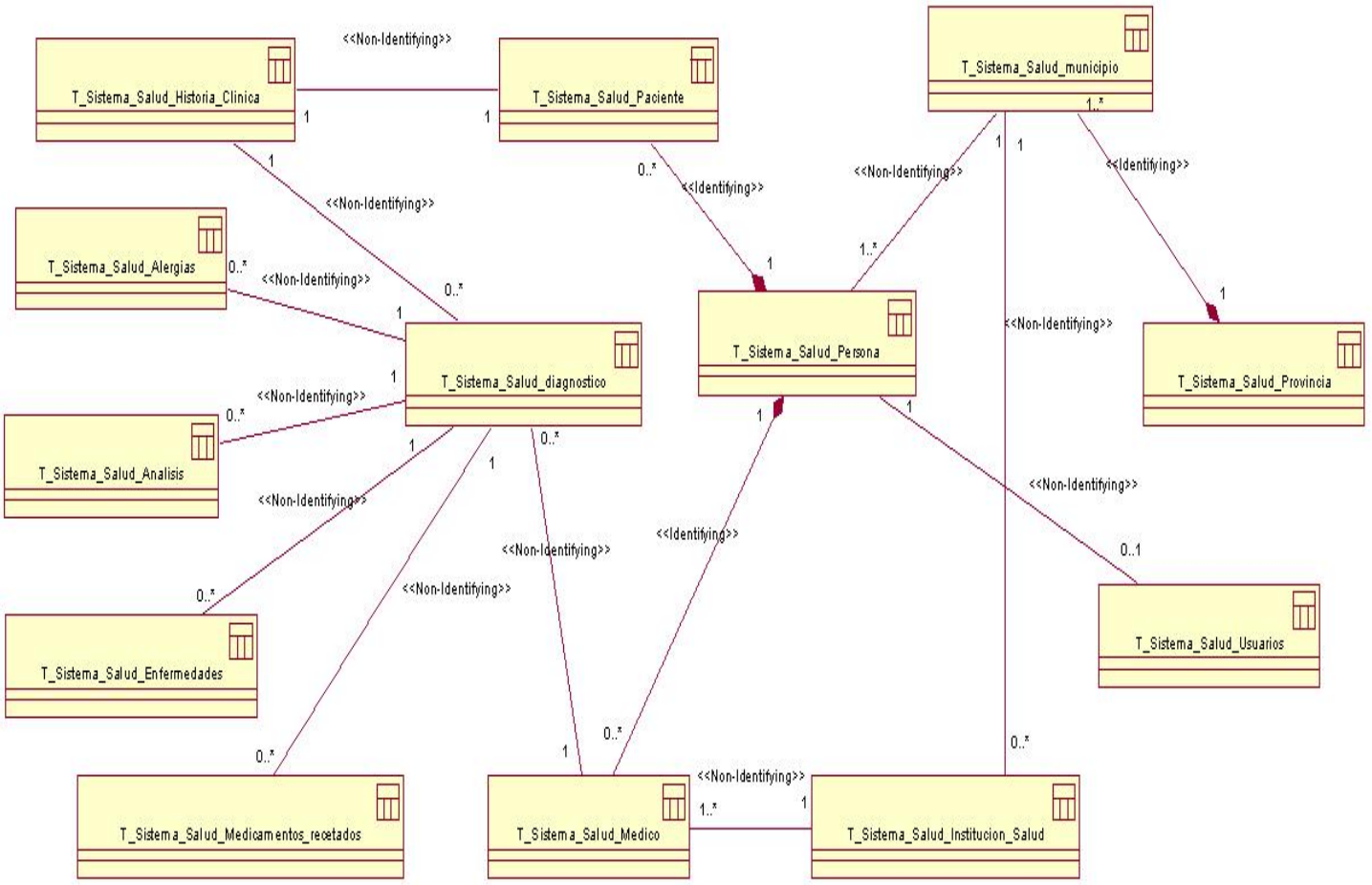


Figura 3.4: Modelo lógico de la Base de datos.

3.5.1 Descripción de las clases.

Nombre: T_SistemaSalud_Historia_Clinica		
Descripción: Almacena los datos que conforman la Historia Clínica del paciente		
Atributo	**Tipo**	**Descripción**
Cod_hc	SMALLINT	Identificador de la HC
Fecha_actualizacion	DATETIME	Fecha de la ultima actualización que se realizo en la HC
Cod_paciente	SMALLINT	Identificador del paciente
Carne_identidad	SMALLINT	Carne identidad del paciente

- Para ver las demás descripciones consultar el [Anexo 27].

3.6 Principios de diseño

3.6.1 Interfaz de usuario

La herramienta computacional diseñada es una aplicación Desktop, por lo que la interfaz diseñada para el sistema está basada en el estándar de ventanas. El tipo de letras a utilizar será *Dialog*, tamaño 12 estilo *Plain.* El diseño de la aplicación debe ser sencillo y simple de utilizar por el tipo de usuario al que va dirigida (Médico). Todos los botones tendrán una altura 20 y el ancho variará de acuerdo al contenido del mismo. El sistema mostrará una barra de menú el la parte superior donde estarán la mayoría de las funciones a realizar.

3.6.2 Tratamiento de errores

En el diseño de la aplicación se tendrá en cuenta el tratamiento de errores, mostrando mensajes de fácil comprensión y que sean lo más descriptivos posibles para una mejor información al usuario. Además se tendrá en cuenta alertarlos de posibles riesgos que puedan ocurrir con las operaciones que el usuario realice. Para los mismos se utilizaran los iconos correspondientes a los estándares, ejemplo:

Figura 3.5: Íconos de mensajes.

3.7 Modelo físico de datos.

Tal como se había dicho anteriormente, la aplicación cuenta con una Base de Datos para facilitar la gestión de la información, el modelo físico de datos propuesto es el que sigue:

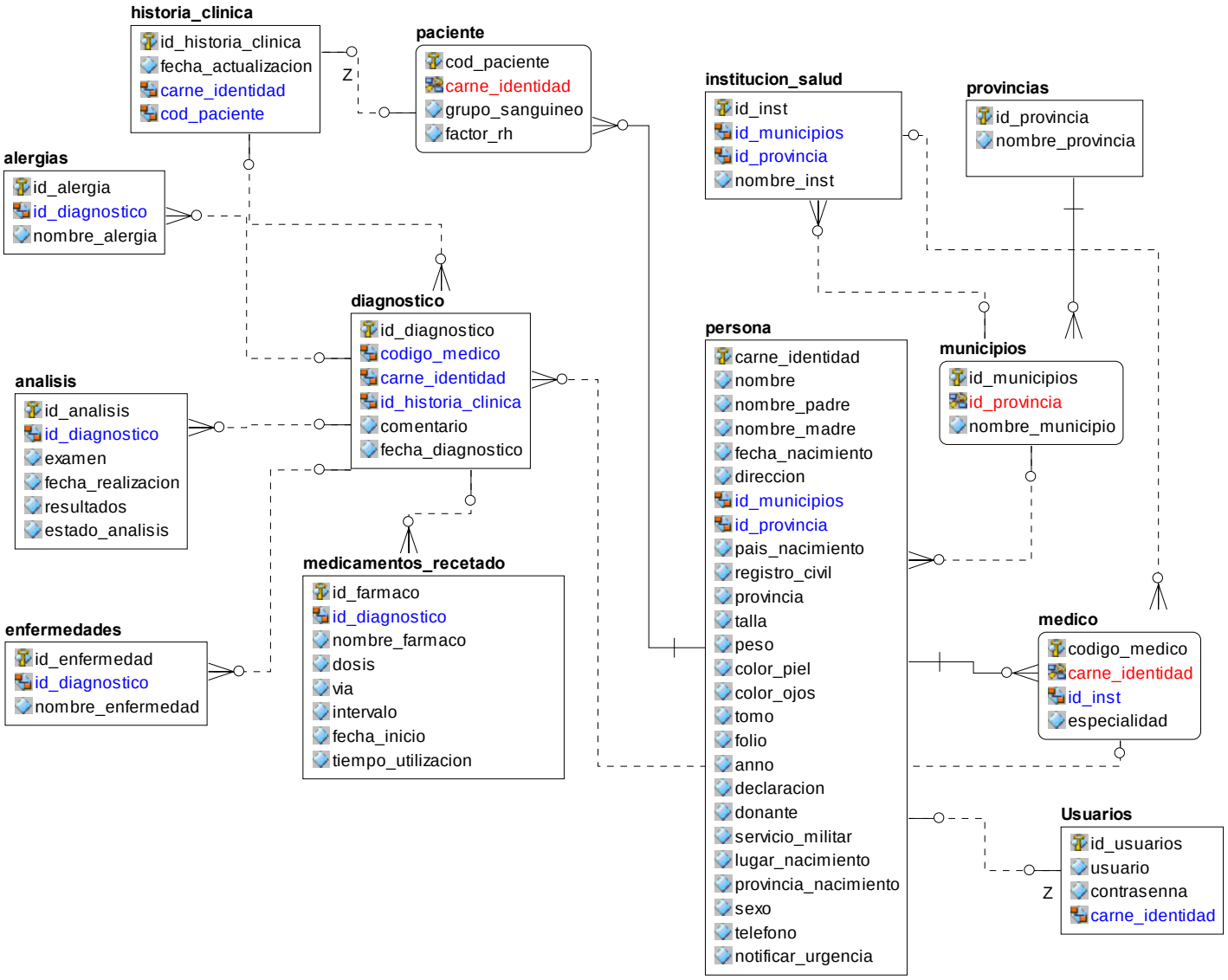


Figura 3.6: Diseño de la Base de Datos.

3.8 Estilo arquitectónico a utilizar.

Arquitectura en capas: Se utilizará la arquitectura en capas debido a que organiza el modelo de diseño a través de capas que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. La cantidad de niveles que se utilizarán serán tres (es decir tres capas). Las mismas se mencionan a continuación:

- Capa de presentación: representa las interface de la aplicación.
- Capa de acceso a datos: representa las clases controladoras y entidades del negocio.
- Capa de datos: representa la base de datos donde esté almacenada la información del sistema.

3.9 Conclusiones.

Con la finalización de este capítulo se da a conocer cómo debe ser implementado el software, partiendo de las clases del diseño definidas, así como también, el diseño que tendrá la BD. Se conoció cómo llevar a cabo las funcionalidades que requieren el sistema que se desarrolló.

CAPÍTULO 4: IMPLEMENTACIÓN

4.1 Introducción.

En este capítulo se describe cómo los elementos del modelo de diseño se implementan en términos de componentes, para esto se muestran los diferentes diagramas de componentes. Además se muestra el diagrama de despliegue con el objetivo de mostrar la distribución física de los nodos de cómputo que necesita la aplicación.

4.2 Diagrama de Despliegue.

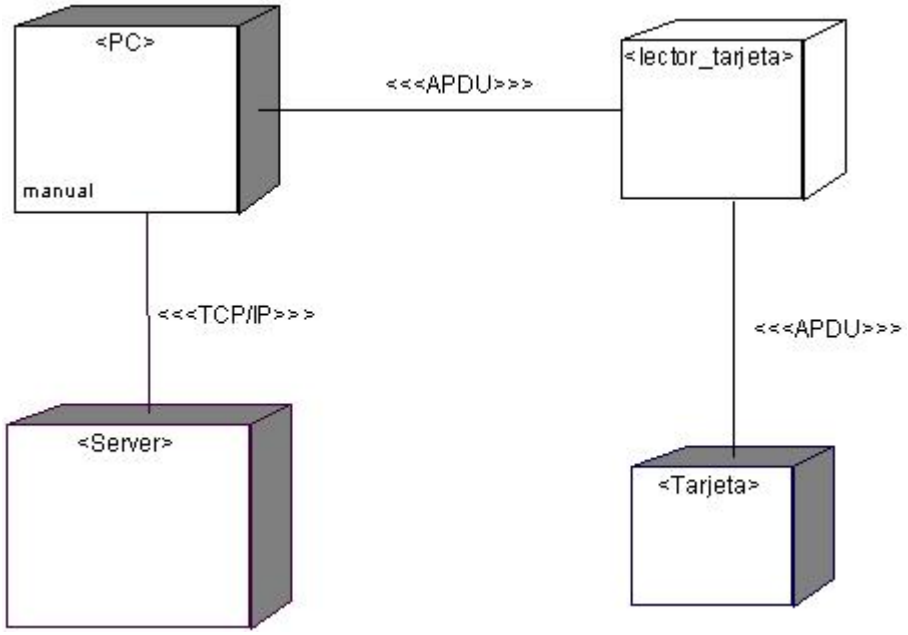


Figura 4.1: Diagrama de Despliegue.

4.3 Diagrama de Componentes.

4.3.1 Clases de Implementación.

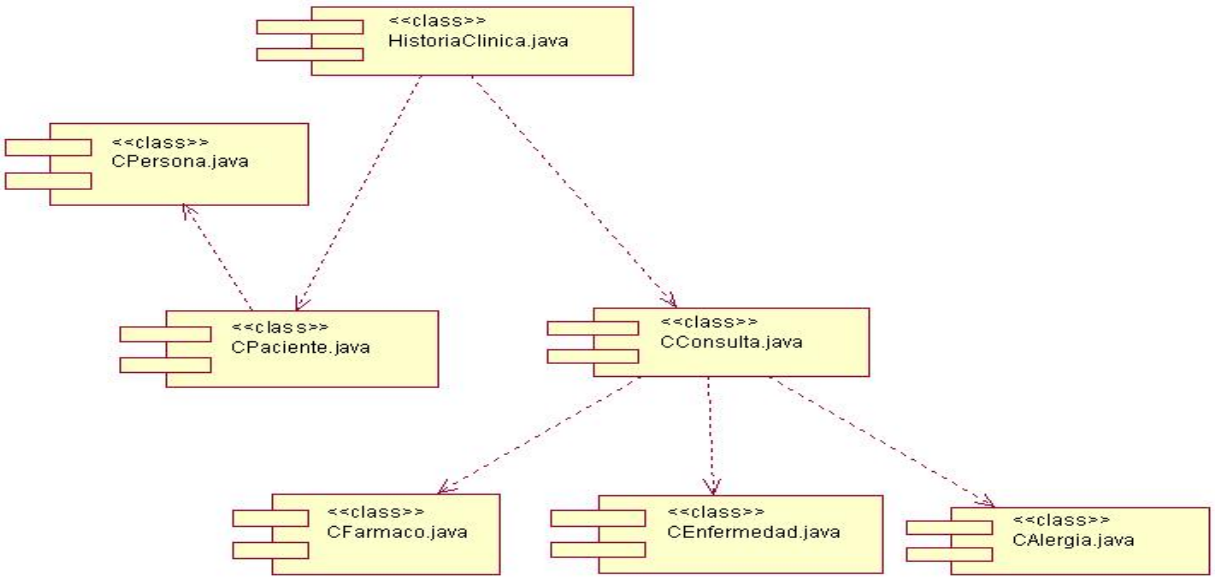


Figura 4.2: Diagrama componentes de las clases de implementación.

4.3.2 Aplicación Salud.

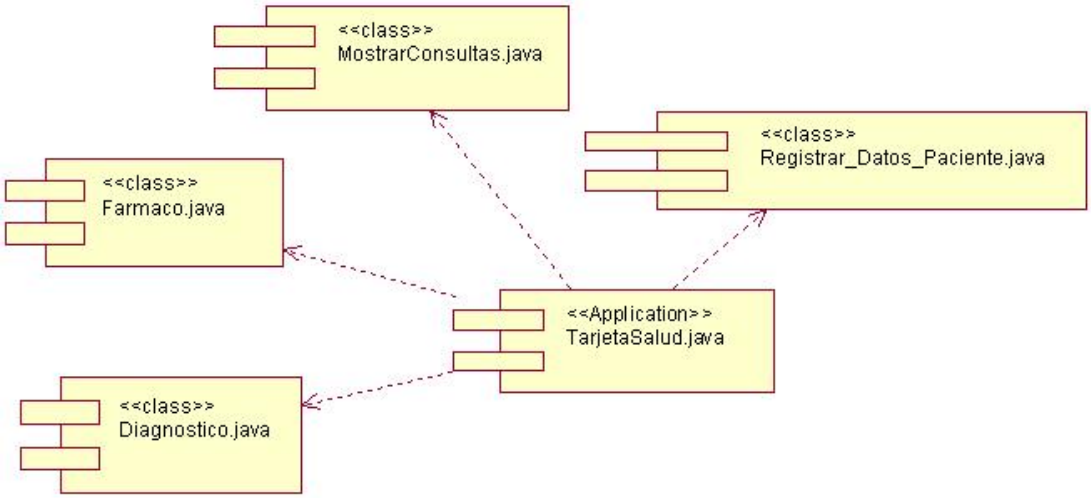


Figura 4.3: Diagrama componente de la aplicación de salud.

4.3.3 Aplicación Identificación.

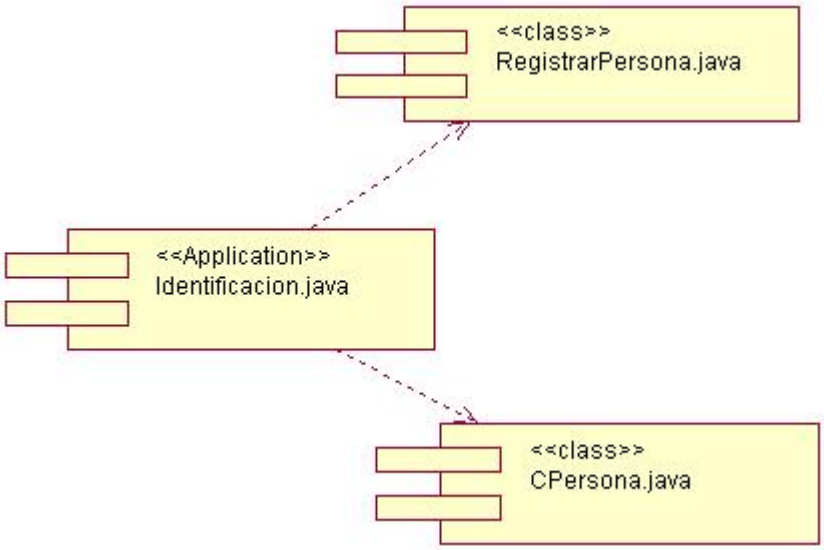


Figura 4.4: Diagrama componentes de la aplicación de identificación.

4.3.4 Aplicación Tarjeta.

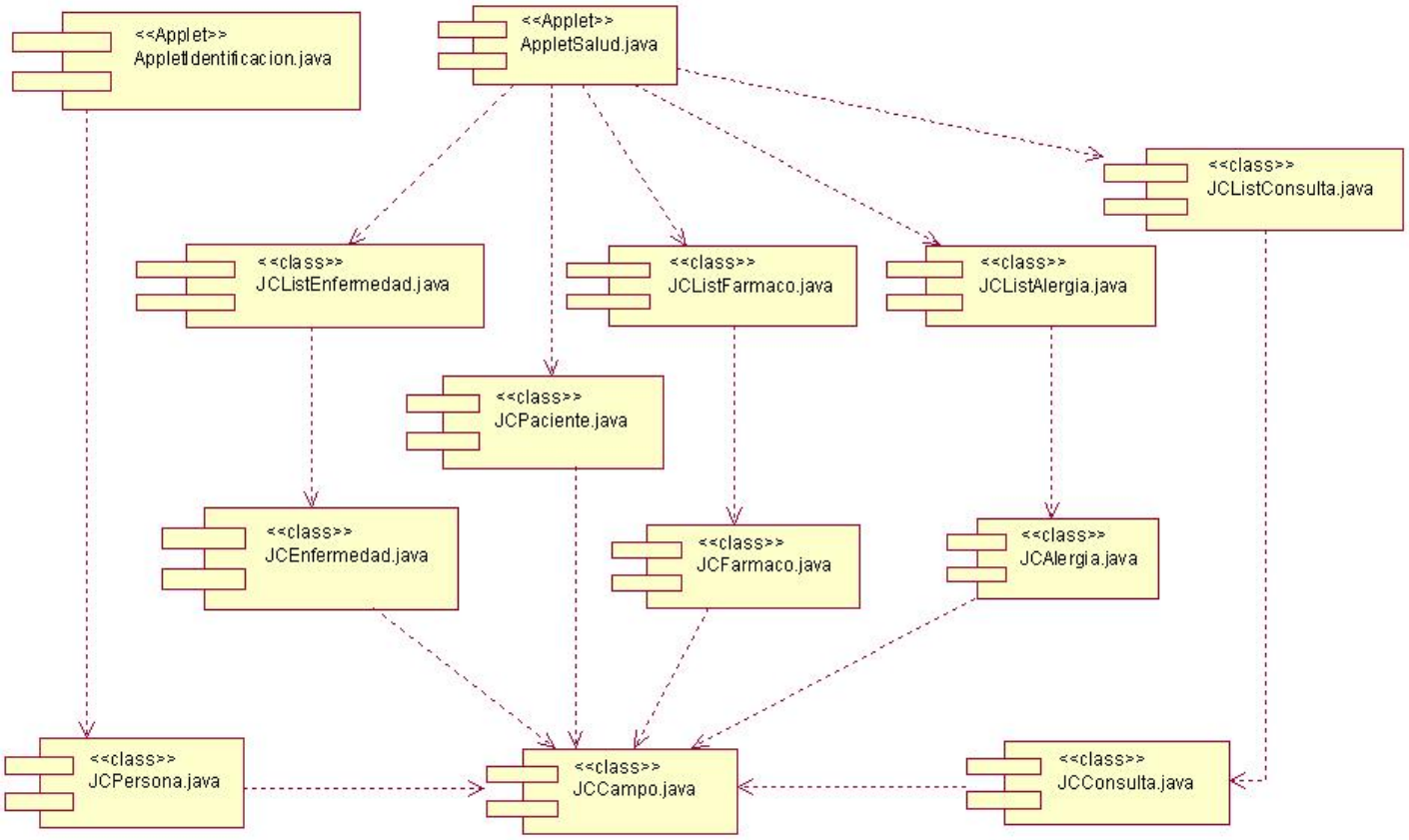


Figura 4.5: Diagrama componentes de la aplicación de la tarjeta.

CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD.

5.1 Introducción.

En este capítulo se realiza una planificación basada en casos de uso con el objetivo de conocer el costo de desarrollar el software.

5.2 Calculo de estimación de costo.

Para la realización de un proyecto es de suma importancia el análisis del costo y los beneficios que reportará. Como resultado de este análisis se obtiene el tiempo de desarrollo en meses, costo y la cantidad de personas que se necesitan para desarrollar el proyecto.

Planificación basada en Casos de Uso.

Paso 1. Cálculo de los Puntos de casos de uso Desajustados.

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de casos de uso sin ajustar.

UAW: Factor de peso de los actores sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar.

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	2	6
Total			2	6

Tabla 4.1 Factor de peso de los actores sin ajustar.

$$UAW = \sum cant\ actores * peso$$

$UAW = 6$

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	6	30
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	0	0
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0
Total			6	30

Tabla 4.2 Factor de peso de los casos de uso sin ajustar.

$$UUCW = \sum cant\,CU * Pesc$$

UUCW = 30

UUCP = 6

UUCP = 36

Paso 2. Cálculo de los Puntos de casos de uso ajustados.

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: Puntos de casos de uso ajustados.

UUCP: Puntos de casos de uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

El factor de complejidad técnica (TCF) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	3	3
T4	Funcionamiento Interno complejo	1	5	5
T5	El código debe ser reutilizable	1	2	2
T6	Facilidad de instalación	0,5	1	0,5
T7	Facilidad de uso	0,5	4	2
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	2	2
T10	Concurrencia	1	0	0
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	1
Total				30,5

Tabla 5.3 Factor de complejidad técnica.

$$TCF = 0.6 + 0.01 * \sum (peso * valor\ asignado)$$

TCF = 0.6 + 0.01 * 30.5

TCF = 0.905

El factor de ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	0	0
E2	Experiencia en la aplicación	0,5	3	1,5
E3	Experiencia en la orientación a objetivos.	1	5	5
E4	Capacidad del analista líder.	0,5	3	1,5
E5	Motivación.	1	5	5
E6	Estabilidad de requerimientos	2	2	4
E7	Personal Part–Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	4	-4
Total				13

Tabla 5.4 Factor de ambiente.

$$EF = 1.4 - 0.03 * \sum (peso * valor\ asignado)$$

EF = 1.4 – 0.03 * 13
EF = 1.01
UCP = UUCP * TCF * EF
UCP = 36 * 0.905 * 1.01
UCP = 32.9058

Paso 3. Estimación de esfuerzo a través de los puntos de casos de uso.

$$E = UCP * CF$$

Donde:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

Para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

En este caso se puede decir que:

CF = 20 Horas-Hombre / Punto de Casos de uso.

E = 32.9058* 20

E = 658.116 Horas-Hombre

Paso 4. Calcular esfuerzo de todo el proyecto.

Actividad	Porcentaje %	Horas-Hombres
Análisis	30	493.587
Diseño	20	329.058
Implementación	40	658.116
Pruebas	10	164.529
Sobrecarga (otras actividades)	0	0
Total	100	1645.29

Tabla 5.5 Esfuerzo del proyecto.

Si **E_T = 1645.29 horas-hombre** y se estima que cada mes tiene como promedio 192 horas laborables, eso daría un **E_T = 8.56921875 mes-hombre.**

Esto quiere decir que 1 persona puede realizar el problema analizado en 7 meses aproximadamente.

-Costo del Proyecto.

Se asume como salario promedio mensual $50.00

CH: Cantidad de hombres.

Tiempo: Tiempo total del proyecto.

CH = 2 hombres

CHM = 2 * Salario Promedio

CHM = 100.00 $/mes

Costo = CHM * E_T / CH

Costo = 100.00 * 8.56921875 / 2

Costo = $ 428.460938 ≈ $428.00 $/h

Tiempo = E_T / CH

Tiempo = 8.56921875 / 2

Tiempo = 4.2846 **≈ 4.28** meses

De los resultados obtenidos se interpreta que con 2 hombres trabajando en el proyecto el mismo se desarrolla en 4 meses y 8 días y con un costo total de $856.00 aproximadamente.

5.3 Beneficios Tangibles e Intangibles.

El beneficio fundamental del sistema es contar con una aplicación Desktop flexible, dinámica y de interfaz agradable que les permita a los médicos cubanos, la gestión de la información de forma rápida y uniforme. Por tanto, los beneficios inmediatos son generalmente intangibles:

- Disminución del tiempo y esfuerzo.
- Disminución de la cantidad de hojas que se utilizan en la realización de las HC, los análisis,…, etc.
- Disminución de los gastos pues resulta menos costoso crear y procesar información digital que copias duras.
- Rápido acceso a la información almacenada.

5.4 Análisis de Costo y Beneficios.

Desarrollar un producto informático cuesta, dinero y esfuerzo. Justificar entonces su desarrollo depende de los beneficios que reportarían su implantación y utilización. Los beneficios pueden ser económicos y de orden social, estos últimos son de tanta importancia como los primeros.

El sistema que se propone esta dirigido al MINSAP para la gestión de la información de las HC de los pacientes cubanos.

Una vez implantado el sistema éste contribuirá a aumentar la eficiencia en la gestión de la información de las HCs de los pacientes, al disminuir el tiempo en el registro, consulta y actualización de la información.

Analizando el costo del proyecto, los numerosos beneficios que reporta, detallados con anterioridad, se puede concluir que su implementación es realmente factible.

5.5 Conclusiones.

En este capítulo se describió el estudio de factibilidad realizado correspondiente al sistema propuesto, teniendo en cuenta el costo estimado y los beneficios que reportará al ser implantado.

La herramienta propuesta reportará beneficios significativos e importantes para la gestión de la información de las HC, que se desea implementar en el Sistema de Salud cubano, al contribuir a mejorar los procesos que se realizan aquí en función de controlar la información de las HC, lo que indica que es factible implementar la herramienta propuesta.

CONCLUSIONES

Se implementó un prototipo de aplicación desarrollado en EclipseJCDE, que permite emular el intercambio de información de una TIJ.

También se desarrolló un prototipo de aplicación que interactúa con el cliente y con un servidor de base de datos MySQL e intercambia información con el emulador de EclipseJCDE de tecnología de Tarjetas Inteligentes Java.

RECOMENDACIONES

- Seguir investigando acerca de esta tecnología para tener un conocimiento más profundo de la misma.
- Investigar más en cuanto a la seguridad que brindan las Tarjetas Inteligentes Java.
- Probar el funcionamiento de la aplicación en el hardware necesario.
- Fomentar el desarrollo de aplicaciones en esta tecnología y su utilización en el país.

REFERENCIA BIBLIOGRÁFICA

1. Perovich, D., L. Rodríguez, and M. Varela, *Programación de JavaCards*. Marzo 2001.
2. AP-ELECTRONICS.com. *¿QUE ES UNA TARJETA INTELIGENTE?* 2007 http://www.ap-electronics.com/tarjetas_inteligentes.asp?op=10.
3. Wikipedia®. *Tarjeta inteligente*. 8 may 2007 http://es.wikipedia.org/wiki/Tarjeta_inteligente
4. alexei. *¿QUÉ ES UNA TARJETA INTELIGENTE?* http://sistemas.dgsca.unam.mx/publica/tarjeta_inteligente/principal.html.
5. Pérez, J.S.O.J.R.S.O.B.R.D.F.J.S.M.Á.R. *Tarjetas Inteligentes*. 20/05/2004 www.dte.us.es/tec_inf/itis/peri_int/trabajos/TarjetasInteligentes.ppt.
6. Esteban, L.P., *EMULADOR DE SAT*. Marzo 2003.
7. Webelectronica.com, *EMULADOR DE TARJETAS TELEFONICAS*.
8. MEDAGLIA, D. *Tarjetas Inteligentes*. 2001 http://www.monografias.com/trabajos16/tarjetas-inteligentes/tarjetas-inteligentes.shtml.
9. Galicía, C.A., S.F. Sandoval, and A.D.H. López. *Tarjetas inteligentes*. 2005 http://cad.cele.unam.mx/~cerealito/html/c482.html.
10. Microsystems, S., *Sun posibilita programa de carné de identidad electrónico para el gobierno de Bélgica*. 18 October 2005.

BIBLIOGRAFÍA.

http://java.sun.com/

http://java.sun.com/products/javacard/

http://www.javaworld.com/javaworld/jw-03-1998/jw-03-javadev.html

http://www.gemplus.com/

http://www.c3po.es/tarjetas_chip.html?PHPSESSID=cb11dc9acf7a05343c070fbd56b5622d

http://sistemas.dgsca.unam.mx/publica/tarjeta_inteligente/principal.html#aspect

http://www.dcc.uchile.cl/~rbaeza/cursos/proyarq/ccastill/informe2.html

http://www.clubse.com.ar/download/pdf/notasrevistas06/nota01.htm

http://es.wikipedia.org/wiki/Tarjeta_inteligente#Historia

http://www.monografias.com/trabajos10/tarin/tarin.shtml#intro

http://www.servired.es/espanol/indexx.htm

http://www.enterate.unam.mx/Articulos/2003/marzo/tarjinte.htm

http://www.kalysis.com/content/modules.php?op=modload&name=FAQ&file=index&myfaq=yes&id_c at=5

http://www.upm.es/laupm/carneupm/infogen.html

http://www.enterate.unam.mx/Articulos/2004/Enero/multiapli.htm

http://pedrochico.fundacionlasemilla.org/

http://www.fing.edu.uy/inco/proyectos/javacard/

ANEXOS

Anexo 1:

Detalles del Caso de Estudio.

Determinar el tamaño de la información.

El tamaño de la información a almacenar presenta las categorías de interés, así como los campos dentro de la categoría. Se indica el tipo de datos y el tamaño de cada uno de los ítems[10]. Presenta la cantidad de ítems necesarios así como el tamaño total. Para reducir el tamaño de alguno de los ítems se utilizó cierta codificación, la cual se indica en la última columna.

Categoría	Campos	Tipo de datos	Tamaño (bytes)	Codificación
Datos Persona	Carné de Identidad	char	11	
	Nombre	char	50	
	Nombre del padre	char	50	
	Nombre de la madre	char	50	
	Fecha de confección	date	8	
	Dirección particular	char	80	
	Código municipio	char	2	01.01 | 01.02 | …
	Código provincia	char	4	01 | 02 | …
	País de nacimiento	char	30	
	Registro civil	char	2	01.01 | 01.02 | …
	Provincia	char	2	01 | 02 | …
	Talla	char	3	
	Peso	char	3	
	Color de la piel	char	1	B, N, M
	Color de los ojos	char	1	N, P, A, V

[10] Ítems: campos de información (nombre, carné,…, etc.).

	Tomo	char	5	
	Folio	char	5	
	Año	char	4	
	Declaración	char	10	
	Donante	char	1	S, N
	Servicio militar	char	10	INFA-6
	Lugar de nacimiento	char	4	
	Provincia de nacimiento	char	4	
	Teléfono	char	10	
	Notificar en caso de urgencia	char	50	
Subtotal para esta categoría			400	
Datos paciente	Grupo sanguíneo	char	3	
	Factor RH	char	10	
Subtotal para esta categoría			13	
Historia clínica	Código HC	char	4	
	Fecha actualización	date	8	
Subtotal para esta categoría			12	
Alergias	Nombre de la alergia	char	50	
Subtotal para esta categoría (10 entradas)			500	
Diagnósticos realizados	Código del diagnostico	char	3	
	Código del medico que lo realizó	char	3	

	Comentario	char	100	
	Fecha de realización	date	8	
Subtotal para esta categoría (10 entradas)			1140	
Análisis	Código del análisis	char	3	
	Código del diagnóstico	char	3	
	Examen	char	50	
	Fecha de realización	date	8	
	Resultados	char	50	
	Estado del análisis	char	1	
Subtotal para esta categoría (10 entradas)			1150	
Enfermedades	Nombre enfermedad	char	50	
Subtotal para esta categoría (10 entradas)			500	
Fármacos en uso	Nombre o denominación	char	50	
	Dosis	char	100	
	Unidades	char	5	
	Vía	char	10	
	Intervalo	char	25	
	Fecha de inicio	date	8	
	Tiempo de utilización	char	25	
Subtotal para esta categoría (10 entradas)			2230	

Tamaño Total		5945	

El tamaño calculado es de aproximadamente 6 KB en total. El tamaño es relativamente adecuado para ser almacenado en una tarjeta.

Notar, sin embargo, las limitaciones en los tamaños de cada campo, que lleva a desperdiciar espacio en algunos casos y a quedar escaso en otros. Un ejemplo claro es el campo nombre, donde la mayoría de los nombres tienen menos de 50 caracteres, pero hay casos en que más caracteres son necesarios.

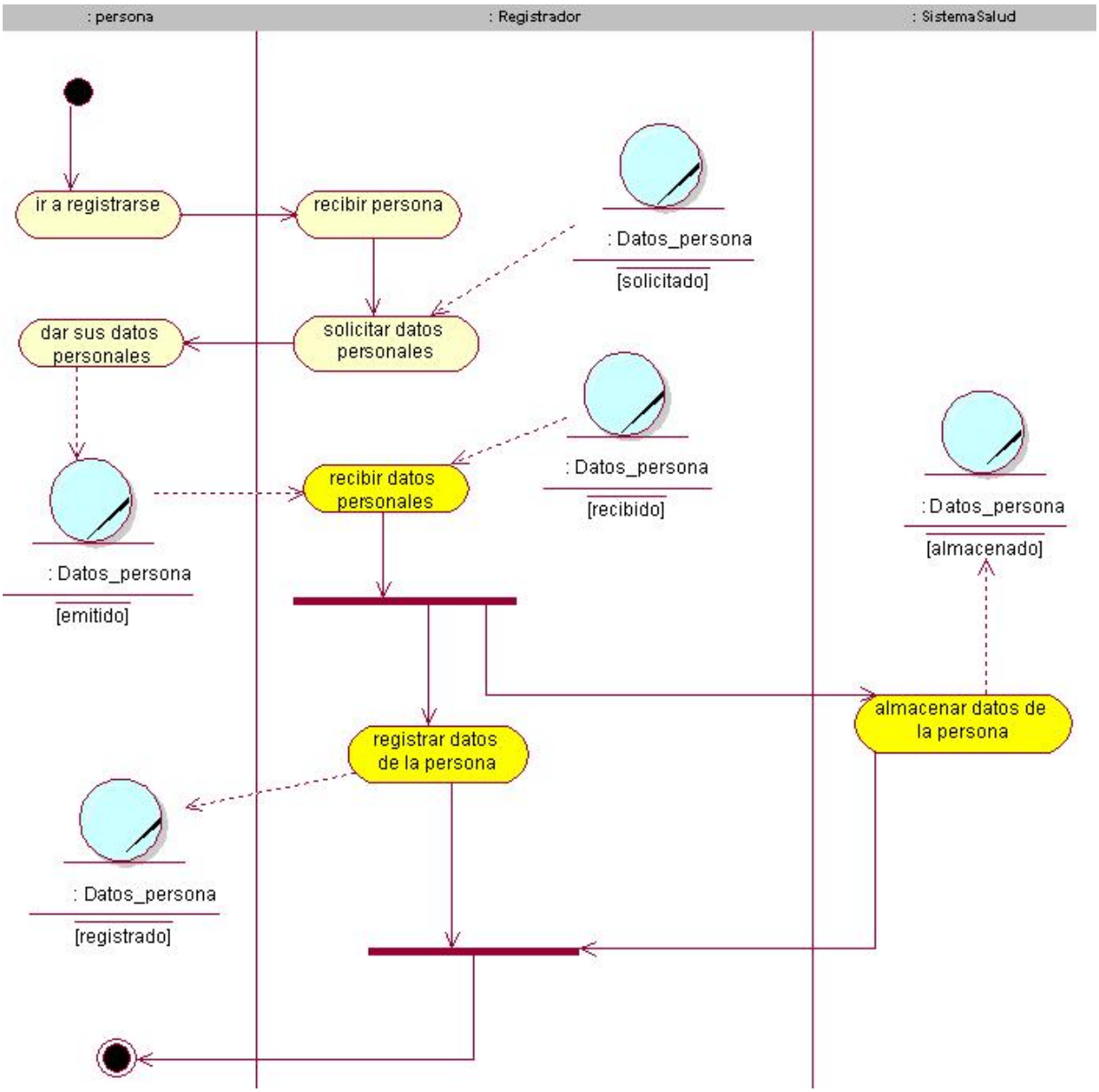
Otro inconveniente es que se asume 10 entradas en las categorías donde se aceptan más de una entrada. Las mismas consideraciones que antes se aplican aquí.

Anexo 2: Descripción del Caso de uso del Negocio Registrar Datos de la Persona.

CASO DE USO DEL NEGOCIO		**Registrar Datos de la Persona.**	
ACTORES	Persona		
PROPÓSITO	Registrar los datos de la persona en la Tarjeta.		
RESUMEN: El caso de uso se inicia cuando la persona llega al local de personalización de la tarjeta, el registrador atiende a la persona y le pide sus datos personales para personalizar la Tarjeta, la persona le entrega sus datos, el registrador recibe los datos y los envía al servidor (SistemaSalud) para almacenarlos, y personaliza la Tarjeta. El caso de uso concluye cuando la persona se retira.			
Precondiciones:	El médico debe estar autenticado.		
CURSO NORMAL DE LOS EVENTOS.			
ACCIÓN DEL ACTOR		**RESPUESTA DEL PROCESO DE NEGOCIO**	
1	La persona llega al local de personalización de la tarjeta.	2	-El registrador recibe a la persona
		3	-El registrador pregunta los datos personales.
4	La persona entrega sus datos personales	5	-El registrador recibe los datos personales.
		6	- El registrador almacena los datos de la persona en el

			servidor (SistemaSalud).
		7	- El registrador personaliza la Tarjeta.
8	La persona se retira del local.		
Prioridad	Crítico		
Mejoras			
Otras secciones			

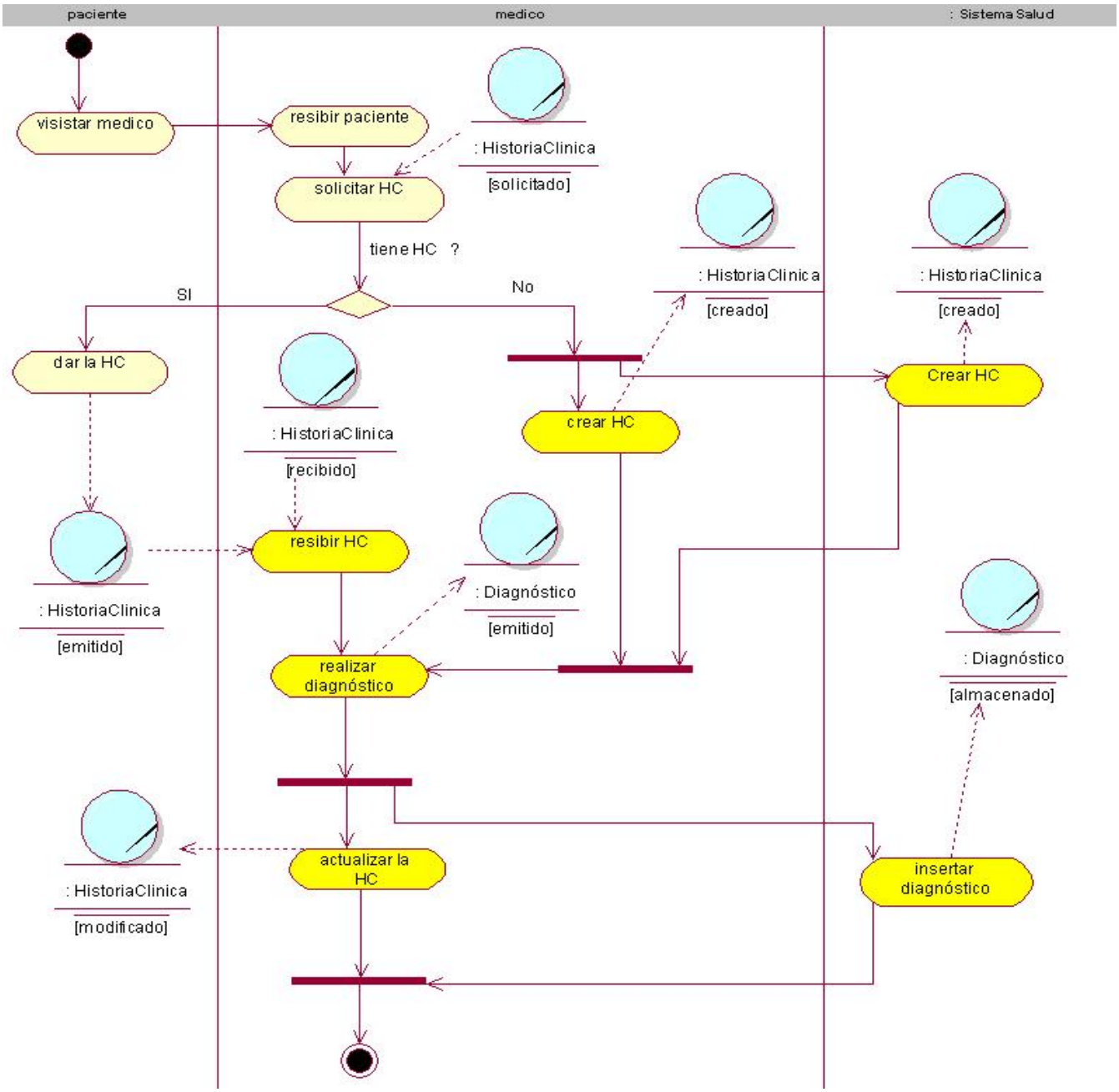
Anexo 3: Diagrama Actividad del Caso de Uso del Negocio Registrar Datos de la Persona.



Anexo 4: Descripción del Caso de uso del Negocio Atender Paciente.

CASO DE USO DEL NEGOCIO		**Atender Paciente.**	
ACTORES	Paciente		
PROPÓSITO	Atender al paciente.		
RESUMEN: El caso de uso se inicia cuando el paciente llega a la consulta, el médico recibe al paciente, le pide su HC, el paciente entrega su HC al Médico, este recibe la HC y realiza el diagnóstico al paciente, luego actualiza la HC y almacena el diagnostico en el servidor (SistemaSalud), con los datos tomados de la consulta. El caso de uso concluye cuando el paciente se retira de la consulta.			
Precondiciones:	El médico debe estar autenticado.		
CURSO NORMAL DE LOS EVENTOS.			
ACCIÓN DEL ACTOR		**RESPUESTA DEL PROCESO DE NEGOCIO**	
1	- El paciente llega a la consulta.	2	-El médico recibe al paciente
		3	-El médico le pide la HC al paciente.
4	- El paciente le entrega su HC	5	-El médico recibe la HC del paciente.
		6	- El médico comienza a realizar el diagnóstico.
		7	- El médico actualiza la HC e inserta el diagnostico en el Servidor (SistemaSalud).
CURSO ALTERNO			
LINEA 4			
ACCIÓN DEL ACTOR		**RESPUESTA DEL PROCESO DE NEGOCIO**	
4	- El paciente informa que no tiene HC.	5	- El médico le crea una HC al paciente.
Prioridad	Crítico		
Mejoras			
Otras secciones			

Anexo 5: Diagrama Actividad del Caso de Uso del Negocio Atender Paciente.



Anexo 6: Descripción del Caso de uso del Sistema Autenticar.

CASO DE USO DEL SISTEMA		**Autenticar.**	
ACTORES	Usuario		
PROPÓSITO	Controlar el acceso a la aplicación.		
RESUMEN: El caso de uso se inicia cuando el Usuario necesita acceder a la aplicación, una vez ejecutada la aplicación, el Usuario deberá introducir su Usuario y Contraseña en el Sistema, en caso de que los datos no sean correctos, la aplicación le mostrará un mensaje de error, y le volverá a pedir los datos, en caso de que los datos no tengan errores la aplicación le permitirá, avanzar al siguiente nivel, concluyendo de esta forma el caso de uso.			
CURSO NORMAL DE LOS EVENTOS.			
ACCIÓN DEL ACTOR		**RESPUESTA DEL SISTEMA**	
1	- El Usuario ejecuta la aplicación para su uso.	1.1	- El Sistema muestra una ventana de autenticación.
2	- El Usuario introduce su Usuario y Contraseña.	2.1	- El Sistema procesa la información, para verificar que los datos sean correctos.
		2.2	- De ser correctos los datos el Sistema muestra el siguiente nivel de la aplicación.
CURSO ALTERNO			
ACCIÓN DEL ACTOR		**RESPUESTA DEL SISTEMA**	
		2.2	- De no ser correctos los datos, el sistema mostrará un mensaje de error advirtiéndole al Usuario que sus datos no son correctos, volviendo a la acción 1.2.

Prioridad	Secundario
Mejoras	
Otras secciones	

Anexo 7: Descripción del Caso de uso del Sistema Registrar Datos Persona.

Caso de Uso del Sistema		Registrar Datos Persona	
Actor	Registrador		
Propósito	Registrar los datos de la Persona.		
Resumen: El caso de uso se inicia cuando el Registrador entra los datos de la Persona y procede a personalizar la Tarjeta, el sistema almacena los datos en la tarjeta, terminando de esta forma el caso de uso.			
Curso normal de los eventos.			
ACCIÓN DEL ACTOR		**RESPUESTA DEL SISTEMA**	
1	- El Registrador introduce los datos de la persona.	1.1	- El Sistema almacena los datos en la tarjeta.
Prioridad		Crítico	
Mejoras			
Otras secciones			

Anexo 8: Descripción del Caso de uso del Sistema Crear HC.

<table>
<tr><th colspan="2">CASO DE USO DEL SISTEMA</th><th colspan="2">Crear HC.</th></tr>
<tr><td colspan="2">ACTORES</td><td colspan="2">Médico</td></tr>
<tr><td colspan="2">PROPÓSITO</td><td colspan="2">Crear la HC del paciente.</td></tr>
<tr><td colspan="4">RESUMEN: El caso de uso se inicia cuando el Médico procede a crear la HC del paciente, el sistema busca los datos de la persona almacenados en la tarjeta y los muestra al médico, entonces este entra los datos del paciente y procede a crearle una HC, el sistema almacena los datos, terminando de esta forma el caso de uso.</td></tr>
<tr><td colspan="2">PRECONDICIONES</td><td colspan="2">Tener una tarjeta y una conexión a la Base de Datos del Sistema y estar autententicado previamente.</td></tr>
<tr><td colspan="2">POSCONDICIONES</td><td colspan="2">Almacenar los datos en la Base de Datos del Sistema.</td></tr>
<tr><td colspan="4">CURSO NORMAL DE LOS EVENTOS.</td></tr>
<tr><td colspan="2">ACCIÓN DEL ACTOR</td><td colspan="2">RESPUESTA DEL SISTEMA</td></tr>
<tr><td>1</td><td>- El médico procede a crear la HC del paciente.</td><td>1.1</td><td>- El Sistema busca los datos del paciente almacenados en la tarjeta y los muestra al médico.</td></tr>
<tr><td>2</td><td>- El Médico introduce los datos del paciente.</td><td>2.1</td><td>- El sistema almacena los datos del paciente en la tarjeta, creando de esta forma la HC del paciente.</td></tr>
<tr><td colspan="2">Prioridad</td><td colspan="2">Crítico</td></tr>
</table>

Anexo 9: Descripción del Caso de Uso del Sistema Realizar Consulta.

<table>
<tr><th colspan="2">CASO DE USO</th><th colspan="2">Realizar Consulta.</th></tr>
<tr><td colspan="2">ACTORES</td><td colspan="2">Médico</td></tr>
<tr><td colspan="2">PROPÓSITO</td><td colspan="2">Realizar consulta</td></tr>
<tr><td colspan="4">RESUMEN: El caso de uso se inicia cuando el médico procede a realizar una consulta al paciente, el sistema obtiene los datos del paciente almacenado en la tarjeta de las consultas anteriores, (ver caso de uso Obtener HC), y se los muestra, el médico introduce los datos obtenidos en la consulta y el sistema almacena los datos en la HC y muestra estos datos al médico, terminando de esta forma el caso de uso.</td></tr>
<tr><td colspan="2">Precondiciones:</td><td colspan="2">El médico debe estar autenticado.</td></tr>
<tr><td colspan="4">CURSO NORMAL DE LOS EVENTOS.</td></tr>
<tr><td colspan="2">ACCIÓN DEL ACTOR</td><td colspan="2">RESPUESTA DEL SISTEMA</td></tr>
<tr><td>1</td><td>- El médico procede a realizarle la consulta al paciente.</td><td>1.1</td><td>- El sistema obtiene los datos del paciente almacenados en la tarjeta de las consultas anteriores y se los muestra. Se invova el caso de uso Obtener HC.</td></tr>
<tr><td>2</td><td>- El Médico introduce los datos del paciente obtenidos en la consulta.</td><td>2.1</td><td>- El sistema almacena los datos de la consulta y los muestra al médico.</td></tr>
<tr><td colspan="2">Prioridad</td><td colspan="2">Crítico</td></tr>
</table>

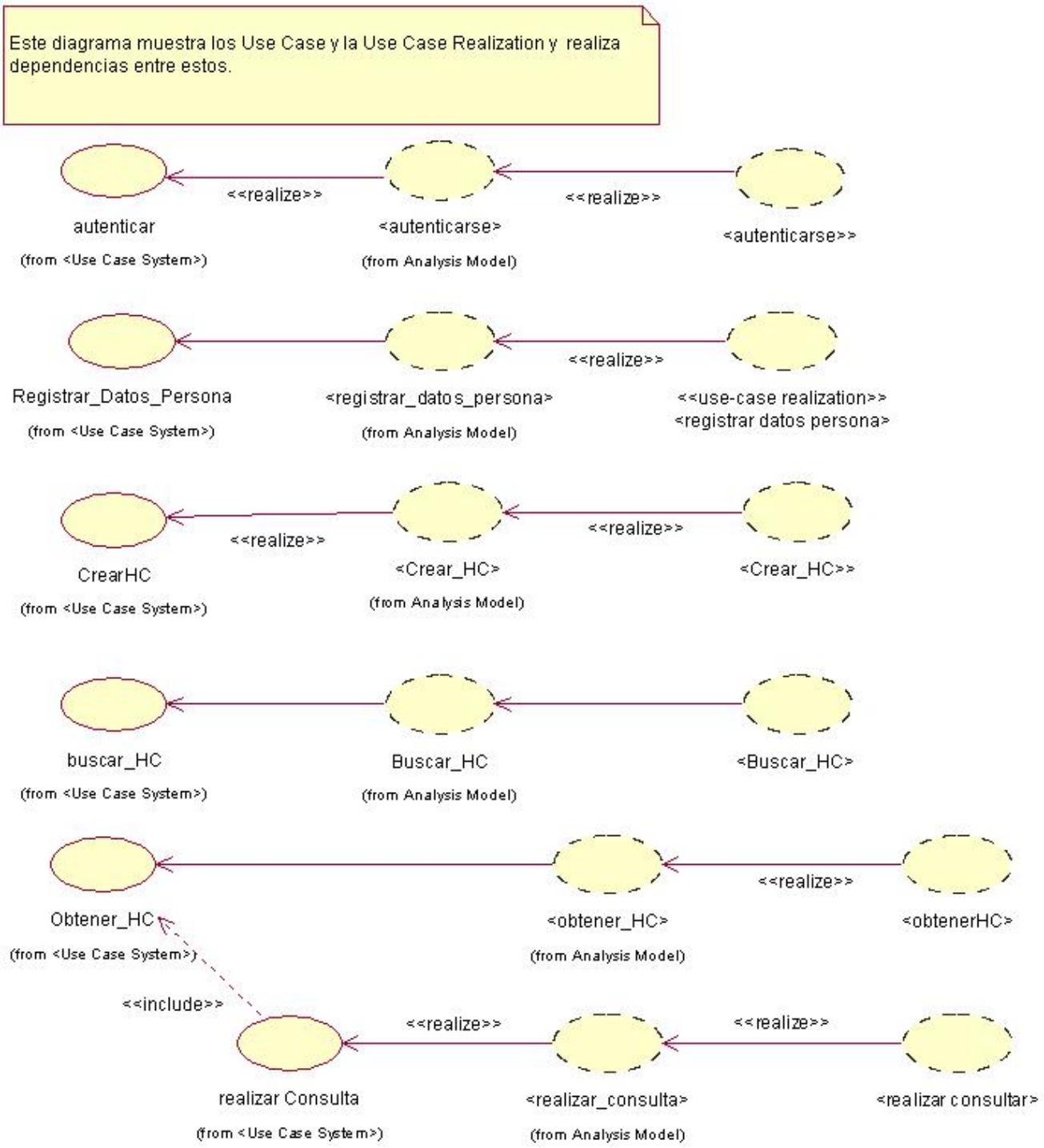
Anexo 10: Descripción del Caso de uso del Sistema Buscar HC.

<table>
<tr><td colspan="2">CASO DE USO</td><td colspan="3">Buscar HC.</td></tr>
<tr><td>ACTORES</td><td colspan="4">Médico</td></tr>
<tr><td>PROPÓSITO</td><td colspan="4">Buscar la HC del paciente.</td></tr>
<tr><td colspan="5">RESUMEN: El caso de uso se inicia cuando el médico procede a buscar la HC del paciente, el sistema obtiene los datos de la persona almacenados en la tarjeta y con esta información busca la HC del paciente, mostrándole al medico la información encontrada, terminando de esta forma el caso de uso.</td></tr>
<tr><td colspan="2">Precondiciones:</td><td colspan="3">El médico debe estar autenticado.</td></tr>
<tr><td colspan="5">CURSO NORMAL DE LOS EVENTOS.</td></tr>
<tr><td colspan="2">ACCIÓN DEL ACTOR</td><td colspan="3">RESPUESTA DEL PROCESO DEL SISTEMA</td></tr>
<tr><td>1</td><td>- El Médico procede a buscar la HC.</td><td>1.1</td><td colspan="2">- El sistema obtiene los datos de la persona almacenados en la tarjeta.</td></tr>
<tr><td></td><td></td><td>1.2</td><td colspan="2">- El sistema busca la HC del paciente y muestra el resultado al médico</td></tr>
<tr><td colspan="2">Prioridad</td><td colspan="3">Crítico.</td></tr>
</table>

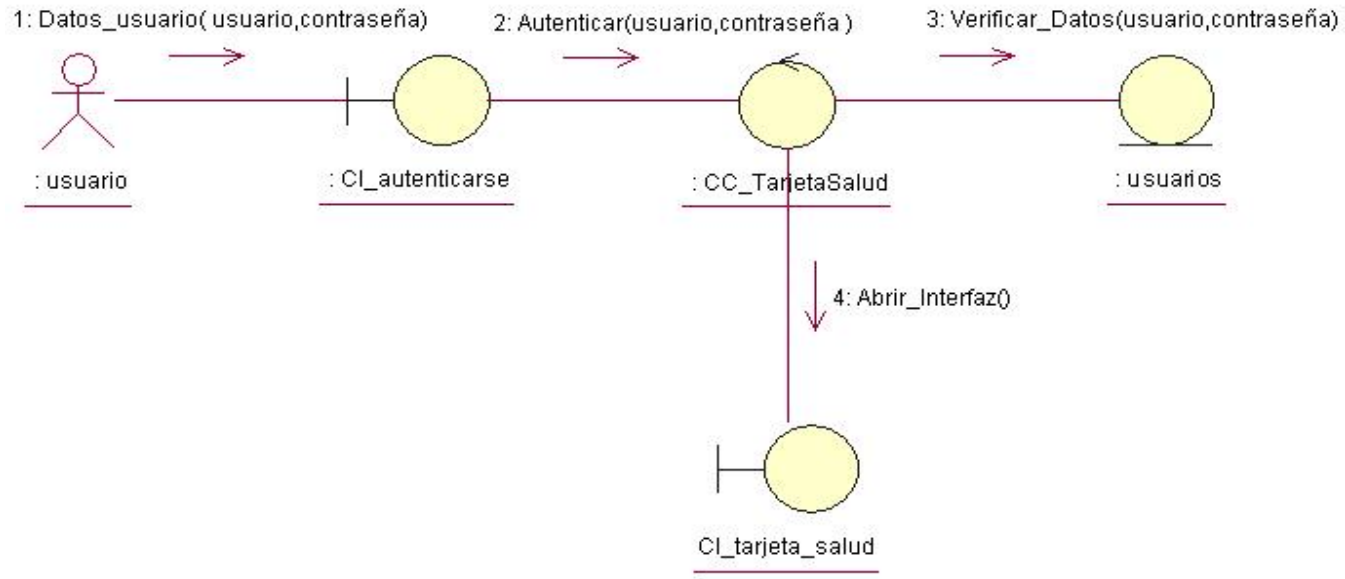
Anexo 11: Descripción del Caso de Uso del Sistema Obtener HC.

CASO DE USO			**Obtener HC.**
ACTORES			
PROPÓSITO	Obtener la HC del paciente contenida dentro de la tarjeta.		
RESUMEN: El caso de uso se inicia cuando el médico procede a realizar la consulta a un paciente, entonces el sistema procede a buscar la HC del paciente que esta contenida dentro de la tarjeta, mostrándole al médico la información encontrada, terminando de esta forma el caso de uso.			
Precondiciones:	El médico debe estar autenticado, y el caso de uso Realizar consulta debe estar inicializado		
CURSO NORMAL DE LOS EVENTOS.			
ACCIÓN DEL ACTOR		**RESPUESTA DEL PROCESO DEL SISTEMA**	
1	- El Médico procede a realizar la consulta.	1.1	- El sistema obtiene los datos de la HC de la persona almacenados en la tarjeta y los muestra al médico.
Prioridad	Crítico.		

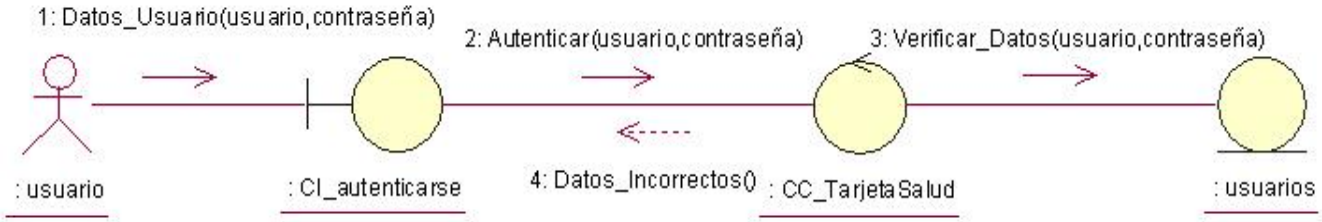
Anexo 12: Realización de los casos de uso del sistema.



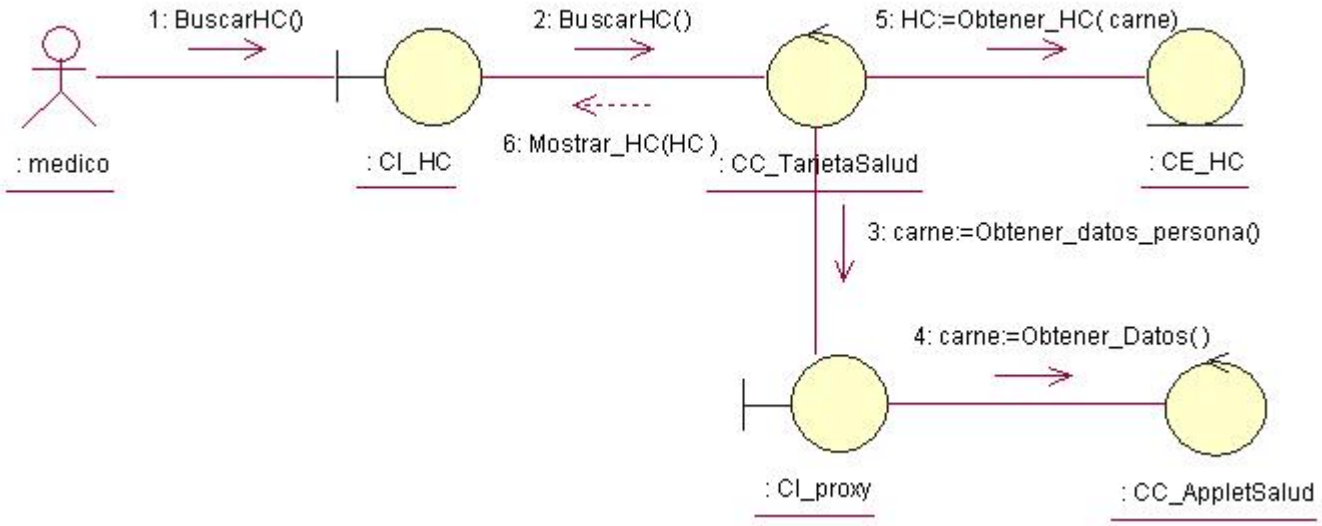
Anexo 13: Diagrama colaboración del análisis Autenticarse.



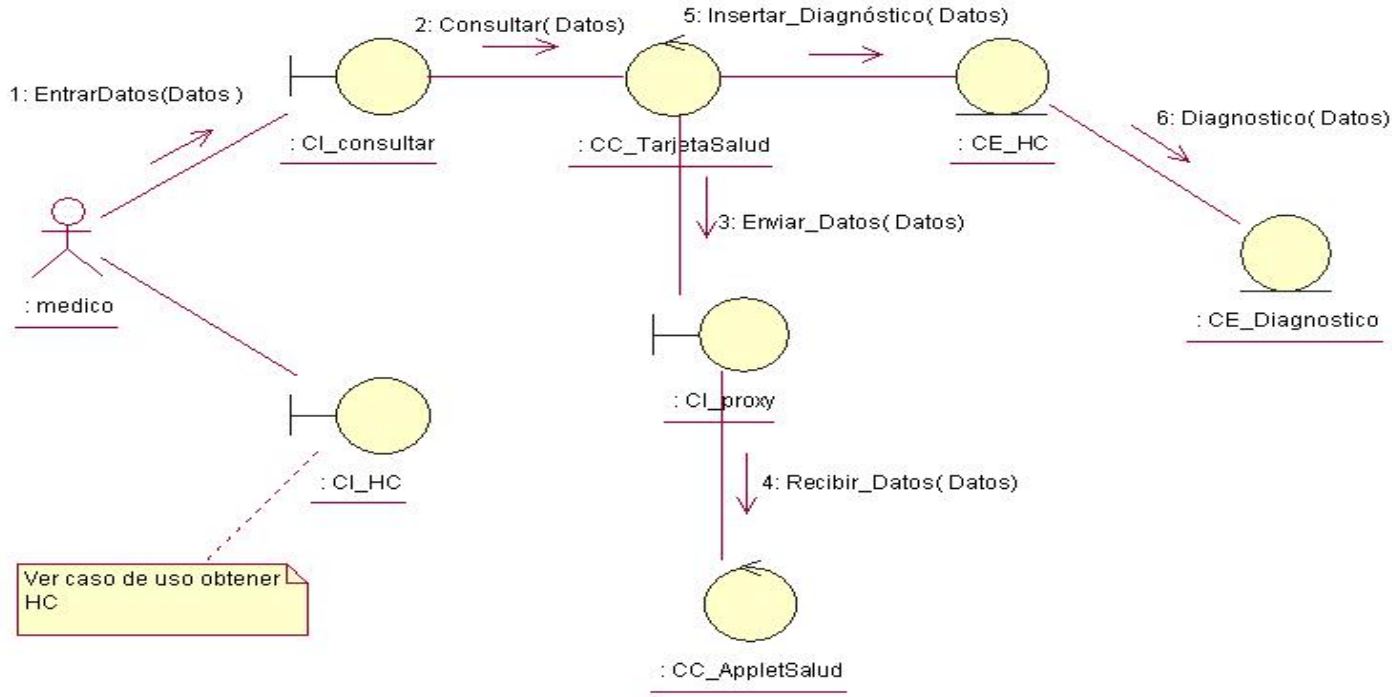
Anexo 14: Diagrama colaboración del análisis Autenticarse. Fallido.



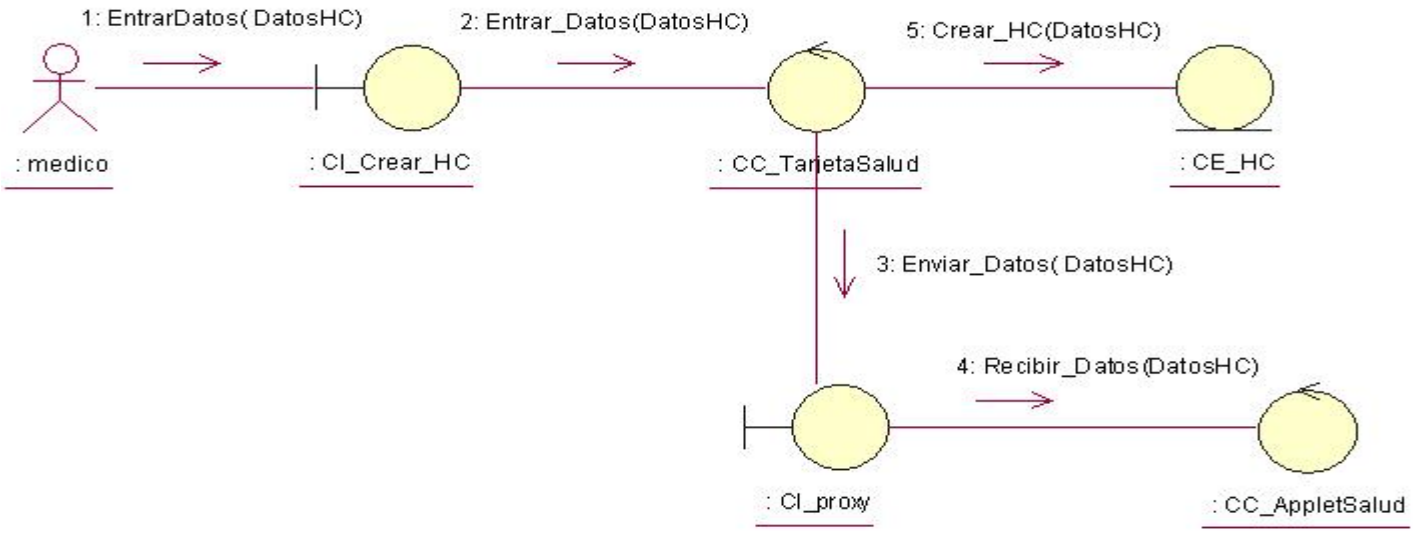
Anexo 15: Diagrama colaboración del análisis Buscar HC.



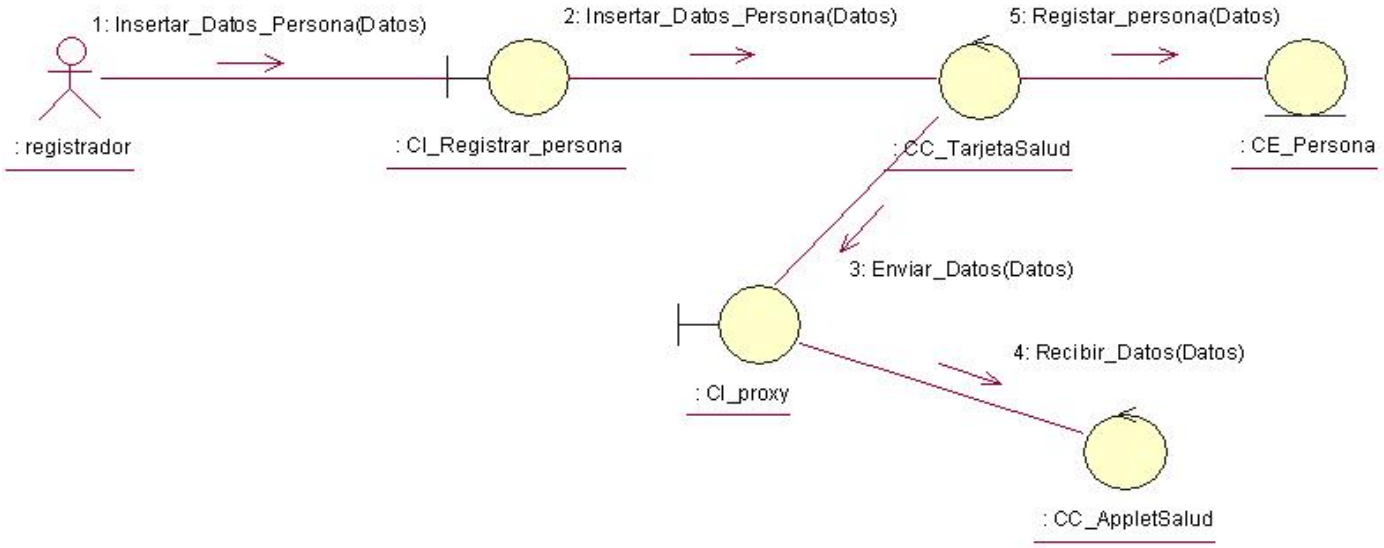
Anexo 16: Diagrama colaboración del análisis Realizar Consulta.



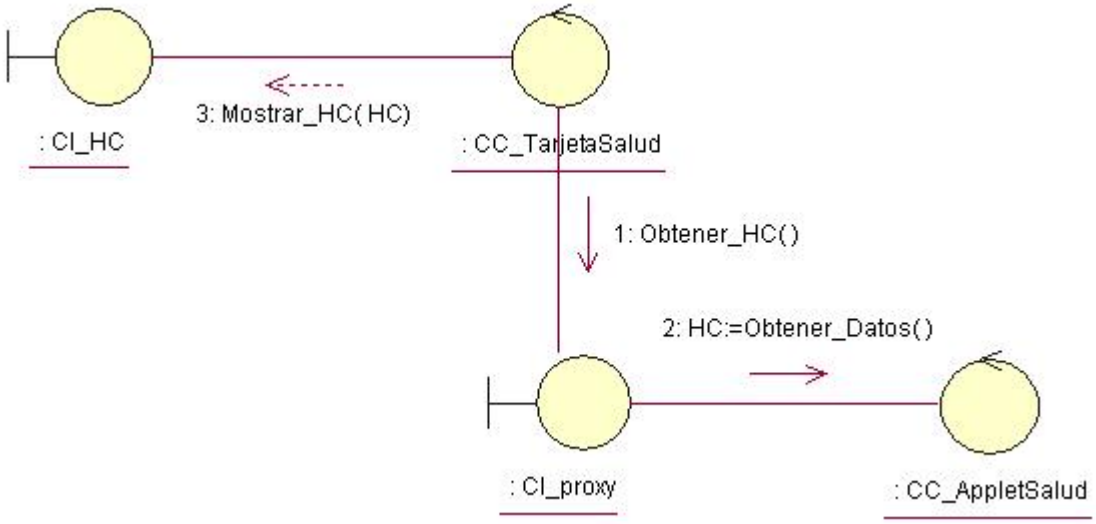
Anexo 17: Diagrama colaboración del análisis Crear HC.



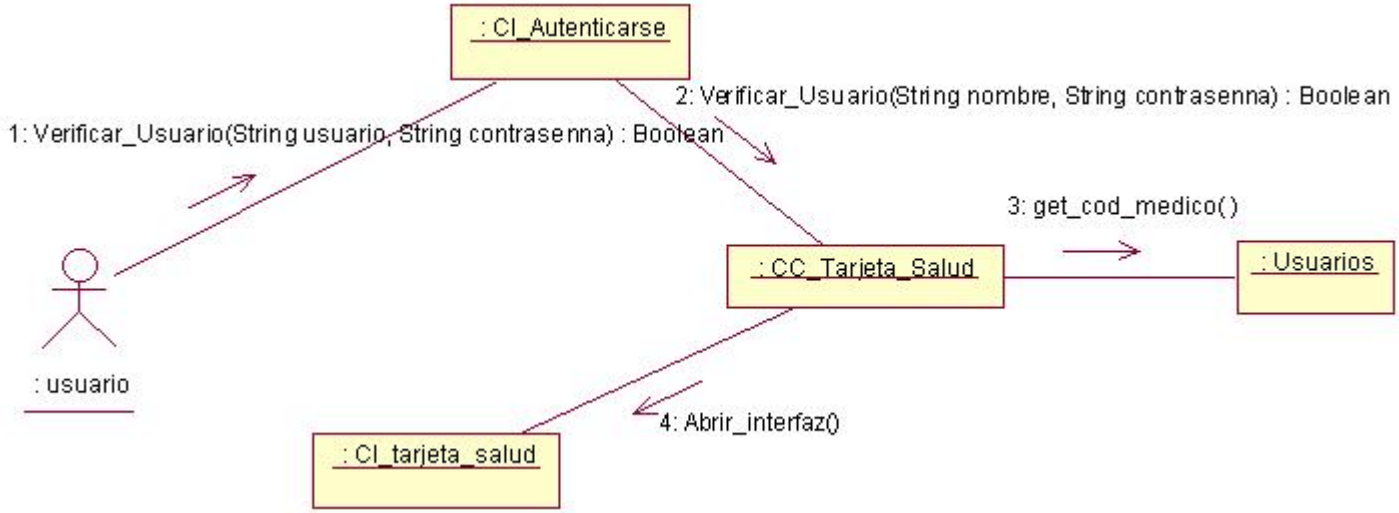
Anexo 18: Diagrama colaboración del análisis Registrar Datos Persona.



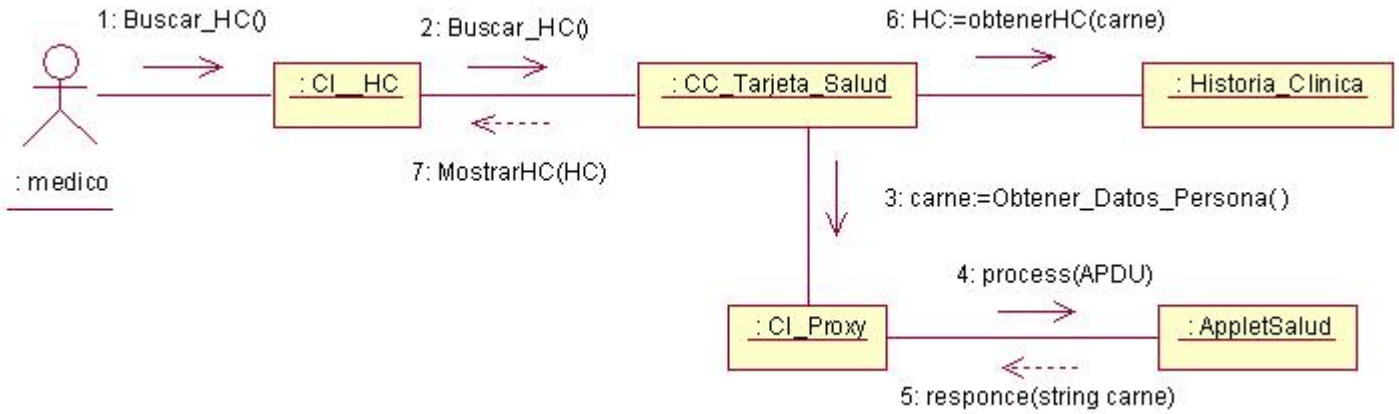
Anexo 19: Diagrama colaboración del análisis Obtener HC.



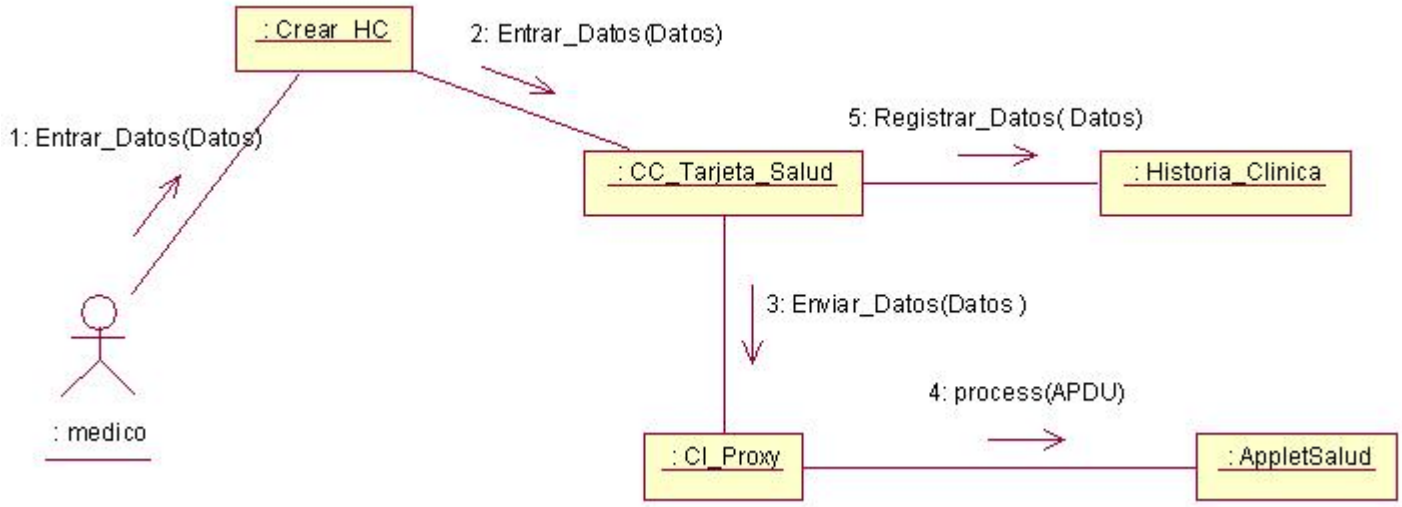
Anexo 20: Diagrama colaboración del diseño Autenticarse.



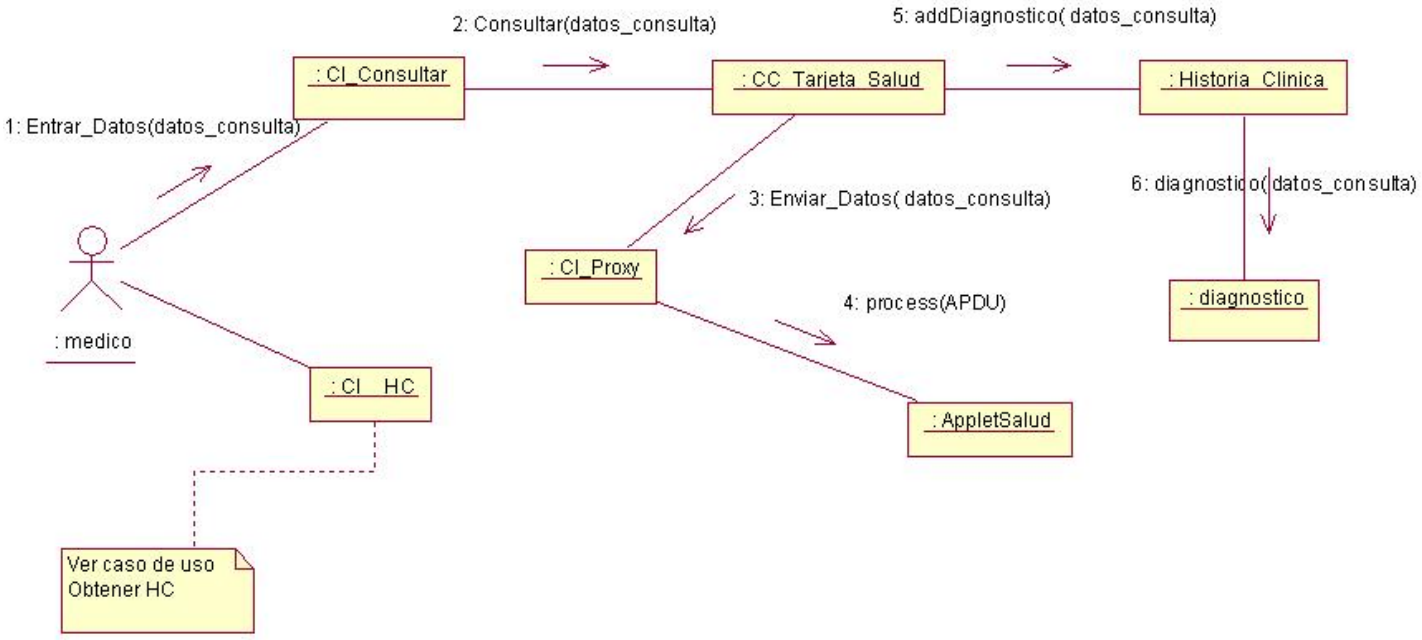
Anexo 21: Diagrama colaboración del diseño Buscar HC.



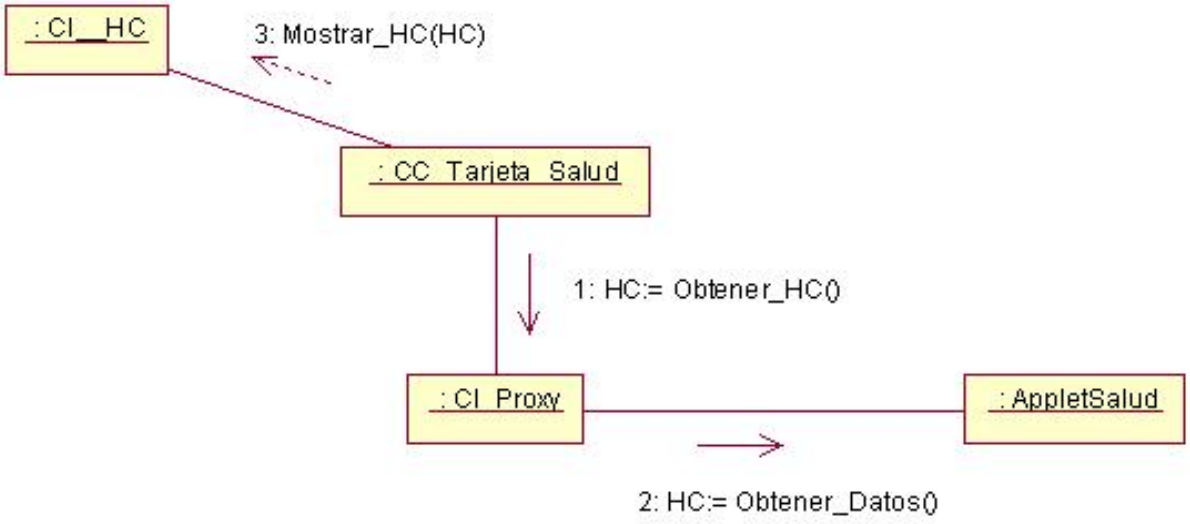
Anexo 22: Diagrama colaboración del diseño Crear HC.



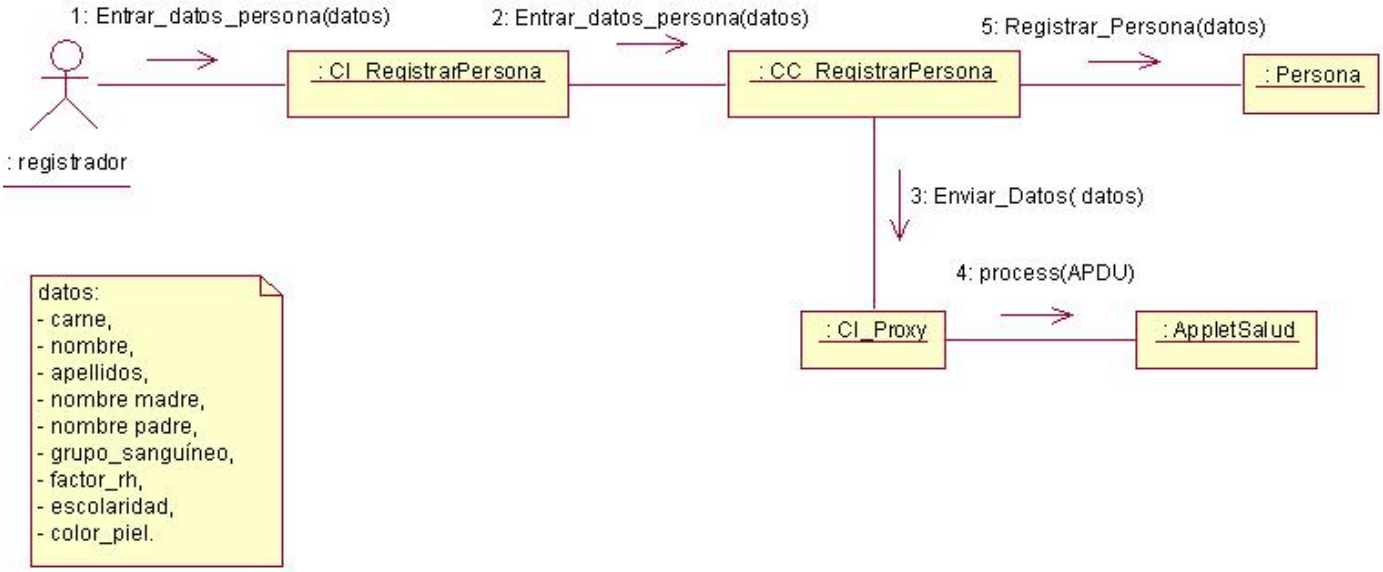
Anexo 23: Diagrama colaboración del diseño Realizar Consulta.



Anexo 24: Diagrama colaboración del diseño Obtener HC.



Anexo 25: Diagrama colaboración del diseño Registrar Datos Persona.



Anexo 26: Descripción de las clases del Diseño.

Nombre:	Alergias
Tipo de clase	entidad
Atributo	**Tipo**
Id_alergia	Char
Nombre_alergia	Char
Para cada responsabilidad:	
Nombre:	Get_id_alergia()
Descripción:	Obtiene el id de la alergia
Nombre:	Get_nombre_alergia()
Descripción:	Obtiene el nombre de la alergia

Nombre:	Historia Clínica
Tipo de clase	entidad
Atributo	**Tipo**
Cod_hc	Char
Fecha_actualizacion	Date
Para cada responsabilidad:	
Nombre:	Get_cod_hc()
Descripción:	Obtiene el código de la HC
Nombre:	Get_fecha_actualizacion()
Descripción:	Obtiene la fecha de la ultima actualización que se realizo en la HC

Nombre:	Diagnóstico
Tipo de clase	entidad
Atributo	**Tipo**
Id_diagnostico	Char
comentario	Char
fecha	Date
Para cada responsabilidad:	
Nombre:	Get_id_diagnostico()
Descripción:	Obtiene el id del diagnóstico
Nombre:	Get_comentario()
Descripción:	Obtiene el comentario plasmado por el medico en el diagnóstico
Nombre:	Get_fecha()
Descripción:	Obtiene la fecha en la que se realizo el diagnóstico

Nombre:	Análisis
Tipo de clase	entidad
Atributo	**Tipo**
Id_analisis	Char
Examen	Char

Fecha_realizacion	Date
Resultados	Char
Estado_analisis	Char
Para cada responsabilidad:	
Nombre:	Get_id_anlisis()
Descripción:	Obtiene el id del análisis
Nombre:	Get_examen()
Descripción:	Obtiene el tipo de examen que se le realizó al paciente
Nombre:	Get_fecha_realizacion()
Descripción:	Obtiene la fecha en la que se realizó el análisis
Nombre:	Get_resultados()
Descripción:	Obtiene los resultados que arrojaron los análisis
Nombre:	Get_estado_analisis()
Descripción:	Obtiene el estado actual en el que se encuentra el análisis (realizado/no realizado)

Nombre:	Medicamento recetados
Tipo de clase	entidad
Atributo	**Tipo**
Id_medicamento	Char
Nombre_medicamento	Char
Dosis	Char
Vía	Char
Intervalo	Char
Fecha_inicio	Date
Tiempo_utilizacion	Char
Para cada responsabilidad:	
Nombre:	Get_id_medicamento()
Descripción:	Obtiene el id del medicamento
Nombre:	Get_nombre_medicamento()
Descripción:	Obtiene el nombre del medicamento
Nombre:	Get_dosis()
Descripción:	Obtiene la dosis del medicamento que se oriento
Nombre:	Get_via()
Descripción:	Obtiene la vía mediante la cual se utilizo el medicamento
Nombre:	Get_intervalo()
Descripción:	Obtiene el intervalo de tiempo en el que se suministro el medicamento
Nombre:	Get_fecha_inicio()
Descripción:	Obtiene la fecha de inicio en la que se suministro el medicamento
Nombre:	Get_tiempo_utilizacion()
Descripción:	Obtiene el tiempo que se estuvo utilizando el medicamento

Nombre:	Enfermedades
Tipo de clase	entidad

Atributo	**Tipo**
Id_enfermedad	Char
Nombre_enfermedad	Char
Para cada responsabilidad:	
Nombre:	Get_id_enfermedad()
Descripción:	Obtiene el id de la enfermedad
Nombre:	Get_nombre_enfermedad()
Descripción:	Obtiene el nombre de la enfermedad

Nombre:	Médico
Tipo de clase	entidad
Atributo	**Tipo**
Id_medico	Char
especialidad	Char
Para cada responsabilidad:	
Nombre:	Get_id_medico()
Descripción:	Obtiene el id del médico
Nombre:	Get_especialidad()
Descripción:	Obtiene el nombre de la especialidad del médico

Nombre:	Paciente
Tipo de clase	entidad
Atributo	**Tipo**
Cod_paciente	Char
Grupo_sanguineo	Char
Factor_rh	char
Para cada responsabilidad:	
Nombre:	Get_cod_paciente()
Descripción:	Obtiene el cod del paciente
Nombre:	Get_grupo_sanguineo()
Descripción:	Obtiene el grupo sanguineo del paciente
Nombre:	Get_factor_rh()
Descripción:	Obtiene el Factor RH del paciente

Nombre:	Persona
Tipo de clase	entidad
Atributo	**Tipo**
Carne_identidad	Char
Nombre_persona	Char
Nombre_padre	Char
Nombre_madre	Char
Fecha_confeccion	Date
Dirección	Char
Pais_nacimiento	Char

Registro_civil	Char
Provincia	Char
Talla	Char
Peso	Char
Color_piel	Char
Color_ojos	Char
Tomo	Char
Folio	Char
Año	Char
Declaración	Char
Donante	Char
Servicio_militar	Char
Lugar_nacimiento	Char
Provincia_nacimiento	Char
Sexo	Char
Teléfono	Char
Notificar_urgencia	Char

Para cada responsabilidad:	
Nombre:	Get_carne_identidad()
Descripción:	Obtiene el carné de identidad de la persona
Nombre:	Get_nombre_persona()
Descripción:	Obtiene el nombre de la persona
Nombre:	Get_nombre_padre()
Descripción:	Obtiene el nombre del padre de la persona en cuestión
Nombre:	Get_nombre_madre()
Descripción:	Obtiene el nombre da la madre de la persona en cuestión
Nombre:	Get_fecha_confeccion()
Descripción:	Obtiene la fecha de confección
Nombre:	Get_direccion()
Descripción:	Obtiene la dirección
Nombre:	Get_pais_nacimiento()
Descripción:	Obtiene el país de nacimiento
Nombre:	Get_registro_civil()
Descripción:	Obtiene el registro civil
Nombre:	Get_provincia()
Descripción:	Obtiene la provincia del registro civil
Nombre:	Get_talla()
Descripción:	Obtiene la talla
Nombre:	Get_peso()
Descripción:	Obtiene el peso de la persona
Nombre:	Get_color_piel()
Descripción:	Obtiene el color de la piel de la persona
Nombre:	Get_color_ojos()
Descripción:	Obtiene el color de los ojos

Nombre:	Get_tomo()
Descripción:	Obtiene el tomo (datos del carne de identidad)
Nombre:	Get_folio()
Descripción:	Obtiene el folio (datos del carne de identidad)
Nombre:	Get_anno()
Descripción:	Obtiene el año (datos del carne de identidad)
Nombre:	Get_declaracion()
Descripción:	Obtiene la declaracion (datos del carne de identidad)
Nombre:	Get_donante()
Descripción:	Obtiene si la persona es donante o no
Nombre:	Get_servicio_militar()
Descripción:	Obtiene el estado de la persona respecto al servicio militar (si lo pasó)
Nombre:	Get_lugar_nacimiento()
Descripción:	Obtiene el lugar de nacimiento de la persona
Nombre:	Get_provincia_nacimiento()
Descripción:	Obtiene la provincia de nacimiento de la persona
Nombre:	Get_sexo()
Descripción:	Obtiene el sexo
Nombre:	Get_telefono()
Descripción:	Obtiene el teléfono
Nombre:	Get_notificar_urgencia()
Descripción:	Obtiene a la persona que se le debe avisar en caso de urgencia

Nombre:	Institución de salud
Tipo de clase	entidad
Atributo	**Tipo**
Id_institucion	Char
Nombre_institucion	Char
Para cada responsabilidad:	
Nombre:	Get_id_institucion()
Descripción:	Obtiene el id de la institución
Nombre:	Get_nombre_institucion()
Descripción:	Obtiene el nombre de la institución

Nombre:	Municipio
Tipo de clase	entidad
Atributo	**Tipo**
Cog_municipio	Char
Nombre_municipio	Char
Para cada responsabilidad:	
Nombre:	Get_cod_municipio()
Descripción:	Obtiene el código del municipio
Nombre:	Get_nombre_municipio()
Descripción:	Obtiene el nombre del municipio

Nombre:	Provincia
Tipo de clase	entidad
Atributo	**Tipo**
Cog_provincia	Char
Nombre_provincia	Char
Para cada responsabilidad:	
Nombre:	Get_cod_provincia()
Descripción:	Obtiene el código de la provincia
Nombre:	Get_nombre_provincia()
Descripción:	Obtiene el nombre de la provincia

Nombre:	Usuarios
Tipo de clase	entidad
Atributo	**Tipo**
Id_usuario	Char
usuario	Char
contraseña	Char
Para cada responsabilidad:	
Nombre:	Get_id_usuario()
Descripción:	Obtiene el id del usuario
Nombre:	Get_usuario()
Descripción:	Obtiene el nombre de usuario

Anexo 27: Descripción de las clases de la Base de Datos.

Nombre: T_SistemaSalud_Historia_Clinica		
Descripción: Almacena los datos que conforman la Historia Clínica del paciente		
Atributo	**Tipo**	**Descripción**
Cod_hc	SMALLINT	Identificador de la HC
Fecha_actualizacion	DATETIME	Fecha de la ultima actualización que se realizo en la HC
Cod_paciente	SMALLINT	Identificador del paciente
Carne_identidad	SMALLINT	Carne identidad del paciente

Nombre: T_SistemaSalud_diagnostico		
Descripción: Almacena los datos que conforman el diagnostico del paciente		
Atributo	**Tipo**	**Descripción**
Id_diagnostico	SMALLINT	Identificador del diagnostico
comentario	SMALLINT	Comentario del médico
fecha	DATETIME	Fecha en la que se realizo el diagnóstico
Cod_medico	SMALLINT	Identificador del medico que realizo el diagnostico
Carne_identidad	SMALLINT	Carne de identidad del medico que realizo el

		diagnostico
Cod_hc	SMALLINT	Identificador de la HC

Nombre: T_SistemaSalud_Analisis		
Descripción: almacena los análisis que se le ha realizado al paciente		
Atributo	**Tipo**	**Descripción**
Id_analisis	SMALLINT	Identificador del análisis
Examen	SMALLINT	Tipo del análisis que se le realizo
Fecha_realizacion	DATETIME	Fecha en la que se le realizo el análisis
Resultados	SMALLINT	Resultados que se obtuvieron con el análisis
Estado_analisis	SMALLINT	Estado en que se encuentra el análisis (enviado, no enviado)
id_diagnostico	SMALLINT	Identificador del diagnostico en el cual se oriento el análisis

Nombre: T_SistemaSalud_Enfermedades		
Descripción: almacena las enfermedades que pose el paciente		
Atributo	**Tipo**	**Descripción**
Id_enfermedad	SMALLINT	Identificador de la enfermedad
Nombre_enfermedad	SMALLINT	Nombre de la enfermedad
id_diagnostico	SMALLINT	Identificador del diagnostico en el cual se oriento el análisis

Nombre: T_SistemaSalud_Medicamentos_recetados		
Descripción: almacena los medicamentos que se le han recetado al paciente		
Atributo	**Tipo**	**Descripción**
Id_medicamendo	SMALLINT	Identificador del medicamento
Nombre_medicamento	SMALLINT	Nombre del medicamento
Dosis	SMALLINT	Dosis que el paciente debe tomar del medicamento
Via	SMALLINT	Vía por la que se proveerá el medicamento
Intervalo	SMALLINT	Intervalo de utilización
Fecha_inicio	DATETIME	Fecha en la que se comienza a utilizar el medicamento
Tiempo_utilizacion	SMALLINT	Tiempo que se debe estar utilizando el medicamento
id_diagnostico	SMALLINT	Identificador del diagnostico en el cual se oriento el análisis

Nombre: T_SistemaSalud_Alergias		
Descripción: almacena las alergias que pose el paciente		
Atributo	**Tipo**	**Descripción**
Id_alergia	SMALLINT	Identificador de la enfermedad
Nombre_alergia	SMALLINT	Nombre de la enfermedad

id_diagnostico	SMALLINT	Identificador del diagnostico en el cual se oriento el análisis

Nombre: T_SistemaSalud_Paciente		
Descripción: almacena datos del paciente		
Atributo	**Tipo**	**Descripción**
Cod_paciente	SMALLINT	Identificador del paciente
Grupo_sanguineo	SMALLINT	Grupo sanguíneo que el paciente posee
Factor_rh	SMALLINT	Factor RH que el paciente posee
Carne_identidad	SMALLINT	Carne de identidad del paciente

Nombre: T_Sistema_SaludPersona		
Descripción: almacena los datos personales de la persona		
Atributo	**Tipo**	**Descripción**
Carne_identidad	SMALLINT	Identificador de la persona
Nombre_persona	SMALLINT	Nombre de la persona
Nombre_padre	SMALLINT	Nombre del padre de la persona
Nombre_madre	SMALLINT	Nombre de la madre de la persona
Fecha_confeccion	DATETIME	Fecha en la que se tomaron los datos de la persona
Dirección	SMALLINT	Dirección particular de la persona
Pais_nacimiento	SMALLINT	País donde nació la persona
Registro_civil	SMALLINT	Estado en el que se encuentra la persona (casado / soltero)
Provincia	SMALLINT	Provincia a la cual pertenece el registro civil
Talla	SMALLINT	Talla de la persona
Peso	SMALLINT	Peso de la persona
Color_piel	SMALLINT	Color de la piel de la persona
Color_ojos	SMALLINT	Color de los ojos de la persona
Tomo	SMALLINT	Datos del carne de identidad
Folio	SMALLINT	Datos del carne de identidad
Año	SMALLINT	Datos del carne de identidad
Declaración	SMALLINT	Datos del carne de identidad
Donante	SMALLINT	Si la persona es o no donante de sus órganos
Servicio_militar	SMALLINT	Si la persona paso o no el servicio militar
Lugar_nacimiento	SMALLINT	Lugar de nacimiento de la persona
Provincia_nacimiento	SMALLINT	Provincia de nacimiento d el apersona
Sexo	SMALLINT	Sexo de la persona
Teléfono	SMALLINT	Teléfono de la persona
Notificar_urgencia	SMALLINT	A quien notificar en caso de una urgencia
Cod_provincia	SMALLINT	Código de la provincia a la cual pertenece
Cod_municipio	SMALLINT	Código del municipio al cual pertenece

Nombre: T_SistemaSalud_Medico

Descripción: Almacena los datos del medico		
Atributo	**Tipo**	**Descripción**
Cod_medico	SMALLINT	Identificador del médico
especialidad	SMALLINT	Especialidad del médico
Id_intitucion	SMALLINT	Identificador de la institución a la cual el médico pertenece
Carne_identidad	SMALLINT	Carné de identidad del médico

Nombre: T_SistemaSalud_Provincia		
Descripción: almacena los datos de la provincia		
Atributo	**Tipo**	**Descripción**
Cod_provincia	SMALLINT	Identificador de la provincia
Nombre_provincia	SMALLINT	Nombre de la provincia

Nombre: T_SistemaSalud_municipio		
Descripción: almacena los datos del municipio		
Atributo	**Tipo**	**Descripción**
Cod_municipio	SMALLINT	Identificador del municipio
Nombre_municipio	SMALLINT	Nombre del municipio
Cod_provincia	SMALLINT	Identificador de la provincia

Nombre: T_SistemaSalud_Institucion_Salud		
Descripción: almacena los datos de la Institución de Salud		
Atributo	**Tipo**	**Descripción**
Id_intitucion	SMALLINT	Identificador de la institución.
Nombre_institucion	SMALLINT	Nombre de la institución de salud
Cod_provincia	SMALLINT	Identificador de la provincia
Cod_municipio	SMALLINT	Identificador del municipio

Nombre: T_SistemaSalud_Usuarios		
Descripción: almacena los datos de los usuarios del Sistema		
Atributo	**Tipo**	**Descripción**
Id_usuario	SMALLINT	Identificador del usuario
usuario	SMALLINT	Nombre del usuario
contrasenna	SMALLINT	Contraseña del usuario
Carne_identidad	SMALLINT	Carne identidad del usuario

1