

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Ingeniería en Computación

Proyecto Terminal

**IMPLEMENTACIÓN DE UNA APLICACIÓN PARA GESTIÓN  
DE LA SALUD PERSONAL CON TARJETAS INTELIGENTES**

Alumno. Godínez Ramírez Angel Akbal  
Matricula. 205202196

Trimestre 100

Asesora. Silva López Rafaela Blanca  
No. Económico.17114

Miércoles 12 de Enero de 2011

## INDICE

OBJETIVO GENERAL	3
OBJETIVOS PARTICULARES	3
INTRODUCCION	4
ANTECEDENTES	4
JUSTIFICACIÓN	6
DESCRIPCIÓN TÉCNICA	7
ESPECIFICACIONES TÉCNICAS	9
ANALISIS Y DISEÑO DE LA BASE DE DATOS RELACIONAL	
• Diccionario de datos	10
• Diagrama Entidad Relación	14
• Esquema Físico	15
DISEÑO E IMPLEMENTACIÓN DE HEALTH CARD MANAGER 2010	
• Diagramas UML de los módulos de HCM 2010	43
• Documentación de actores de HCM 2010	46
• Casos de uso de HCM 2010	47
• Mapa de navegación de HCM 2010	58
• Código fuente de HCM 2010	59
BIBLIOGRAFIA	180

## OBJETIVO GENERAL

Desarrollar una aplicación de software que permita grabar/leer datos personales y de salud en tarjetas inteligentes sin contacto (TISC), a través de un lector/grabador de proximidad para identificación por radiofrecuencia (RFID) para tarjetas TISC (*mifare classic 4k*).

## OBJETIVOS PARTICULARES

- Definir los datos personales y de sanidad, relevantes para el sector salud al atender un paciente.
- Diseñar e implementar una base de datos que permita el respaldo óptimo de la información almacenada de los pacientes registrados.
- Desarrollar el módulo “Control de usuarios”, que registre y controle las altas, bajas y modificación de usuarios en el sistema.
- Crear el módulo “Control de pacientes”, que admita la manipulación y control de datos particulares en la base de datos, para los pacientes registrados en el sistema.
- Realizar el módulo “Datos de sanidad”, que sea capaz de administrar la información de salud para cada paciente. Esta información se almacenará en la base de datos.
- Implementar el módulo “Credencialización”, que permita la asignación de una tarjeta *mifare classic 4k* y la grabación en ella de los datos de salud así como los datos personales, utilizando el lector/grabador de proximidad.
- Desarrollar el módulo “Recolección”, que permitirá extraer y desplegar la información grabada en la tarjeta.
- Implementar la interfaz principal que asegure la integración y funcionamiento de los 5 módulos anteriores.

## INTRODUCCION

En lo que respecta el sector salud a diario se presentan complicaciones al pretender obtener información de salud que sea de vital importancia al atender a los pacientes, información como alergias, enfermedades, discapacidades, tipo de sangre etc.

Como una solución a esta problemática se desarrolló la “Implementación de una aplicación para gestión de la salud personal con tarjetas inteligentes”, dicha aplicación será nombrada “*Health Card Manager 2010*” (HCM 2010).

La aplicación será ejecutada en una computadora personal o de escritorio la cual tendrá como función principal capturar y almacenar en una base de datos información relevante de salud.

HCM 2010 contendrá un módulo capaz de grabar en una tarjeta inteligente sin contacto (TISC) la información de salud del paciente. Para lograr grabar la información de salud almacenada en la base de datos a la tarjeta se hará uso de un Lector/Grabador de identificación por radiofrecuencia (RFID).

De esta manera todas los pacientes podrán portaran consigo su tarjeta y al solicitar alguna revisión médica, todos los datos de salud almacenados en la tarjeta podrán ser extraídos de la misma con ayuda del Lector/Grabador y así el paciente podrá obtener una mejor y pronta atención.

## ANTECEDENTES

Las tarjetas inteligentes han sido de multiuso en todo el mundo, en aplicaciones diversas tales como control de acceso, control de asistencia, pago de transporte masivo, tarjetas de salud, etc.

Una de las aplicaciones con tarjetas inteligentes más conocida, es la implementada con la tarjeta ONA (**Osasun Nortarun Agiria**) [1], en español Documento de Salud e Identidad. Es una Tarjeta Electrónica Sanitaria con usos ciudadanos que emite el Gobierno Vasco [2]. Ofrece servicio de acceso a polideportivos, consulta de catálogos, préstamo de libros en bibliotecas municipales, consulta de datos fiscales y servicios

sanitarios tales como: reserva de cita médica, solicitud de cambio de médico, segunda opinión médica y expedientes asistenciales.

En la Unión Europea (UE) se desarrolló una tarjeta inteligente de salud llamada “Tarjeta Sanitaria Europea” (TSA) [3], con la que cualquier persona procedente de los 27 estados de la UE pueda acceder a los servicios de atención sanitaria durante su estancia temporal en el extranjero.

Con esta tarjeta se asegura el mismo acceso a la atención sanitaria pública que los ciudadanos del país que se esté visitando: se pueden solicitar servicios médicos de un hospital o centro de atención sanitaria.

Por citar un caso, en México, el Seguro Popular desarrolló una tarjeta con chip inteligente, la cual será un documento con validez oficial [4]. Esta tarjeta contará con expedientes clínicos y recetas de los pacientes. Además, se planea poder acceder a otros programas prioritarios, como Oportunidades. A los médicos se les entrega una terminal así como su propia credencial para dar seguimiento a sus actividades. En caso de que el paciente requiera estudios, el médico los registra, también receta las medicinas y las indicaciones para su uso [5].

El sector salud anuncia que se tendrán credenciales electrónicas con historial clínico [6], con lo cual los mexicanos podrán ser atendidos en cualquier sistema de seguridad social. Esta peculiar identificación contendrá todo el historial de vida incluyendo radiografías, enfermedades pasadas y recientes, alergias y cirugías entre otras cosas.

Tomando como referencia estos proyectos que aún se encuentra vigentes, se pretende realizar una tarjeta inteligente, la cual contenga información del portador que sea importante a la hora de recibir servicios médicos. Se desea dividir esta información en dos partes: información de salud general e información de salud confidencial. Para tener acceso a la información de salud confidencial, se solicitará una contraseña de acceso. Esta contraseña será validada con la almacenada en la tarjeta.

Para el acceso a la información de salud general, bastará con tener el módulo que extraerá y desplegará los datos almacenados en la tarjeta, ya que éstos no solicitarán una validación.

En la UAM-A existe un proyecto terminal de ingeniería electrónica en el cual se utilizaron tarjetas inteligentes para el control de acceso de personal [8], este se enfoca más a la creación del lector/grabador de radiofrecuencia.

En el caso de este proyecto nos enfocaremos a la implementación de la aplicación que permita almacenar datos de salud y personales de una persona (paciente) en la tarjeta *mifare classic 4k*.

## **JUSTIFICACIÓN**

En la actualidad hay muchas necesidades en el sector salud cuando surgen emergencias. En algunas ocasiones, los afectados (pacientes) se encuentran inconscientes o conscientes, pero no recuerdan lo sucedido. De ser el caso, resultaría de gran ayuda para los paramédicos que se encuentren con el afectado el tener información de salud que les ayudara a tomar decisiones rápidas para su recuperación.

Por lo tanto, el uso de esta tarjeta de salud que se quiere implementar, sería una alternativa al problema, ya que los paramédicos podrían contar con el módulo para extraer y desplegar la información de la tarjeta, saber de manera inmediata datos importantes como alergias, tipo de sangre y enfermedades del afectado, entre otros, sin necesidad de que éste les proporcione la información.

Es importante mencionar que sin importar si el paciente se encuentra consciente o inconsciente, solo se podrán saber estos datos, siempre y cuando traiga consigo la tarjeta.

No es necesario extraer la tarjeta de la billetera, de la bolsa o de donde la traiga guardada el paciente, ya que las tarjetas que se utilizarán no necesitan contacto; basta con acercar el lector a las zonas posibles donde pueda traerla.

En un futuro sería interesante que esta tarjeta funcionara como monedero electrónico para pagar los medicamentos o estudios que pudiera necesitar el paciente. De igual manera desarrollar el hardware donde se pueda cargar el modulo de extracción de datos de la tarjeta y hacer que este módulo sea independiente de una computadora.

## DESCRIPCIÓN TÉCNICA

Para iniciar el desarrollo de la aplicación se revisó la norma oficial mexicana NOM-168-SSA1-1998 [7], del expediente clínico. Lo anterior permitirá elegir algunos datos relevantes que contendrá la tarjeta de salud.

El diseño e implementación de la base de datos se inició una vez seleccionada la información a almacenar, en el sistema de gestión de base de datos MySQL.

Descripción de los módulos que integran el Sistema:

- Control de usuarios

Al dar por concluida la implementación de la base de datos, se desarrollará el código del módulo que permitirá ingresar y administrar la información de usuarios y sus distintos roles. Este módulo permitirá dar de alta, baja y modificar usuarios que tendrán acceso al sistema.

- Control de pacientes

Este módulo será desarrollado para ingresar y administrar la información personal de los pacientes. Este módulo permitirá dar de alta, baja y modificar pacientes en el sistema.

- Datos de sanidad

Este módulo permitirá administrar la información de salud de cada paciente ya registrado en el sistema. Será capaz de agregar y modificar tipo de sangre, alergias, enfermedades, discapacidades etc.

- Credencialización

El módulo de credencialización permitirá asignar una tarjeta única, en la que se grabarán los datos personales del paciente, así como sus datos de salud. Para lograr grabar la información en la tarjeta se utilizara un lector/grabador ACRU120U.

- Recolección

Permitirá extraer y desplegar la información que se encuentra almacenada en la tarjeta. Será utilizado el lector ACR120U para extraer la información de la tarjeta

- Interfaz de Integración

Realizada la implementación de los módulos anteriores, se procederá a desarrollar la interfaz de integración que permita el acceso a cada uno de ellos.

Plataforma de desarrollo:

- Sistema operativo: Windows XP x86
- Lenguaje de programación: C#
- Entorno de desarrollo: Visual C# Express Edition 2005
- Sistema de gestión de base de datos: MySQL

Plataforma de Ejecución:

- Sistema operativo: Windows XP x32
- Sistema de gestión de base de datos: MySQL

Diagrama a bloques de Health Card Manager 2010. *Figura 1*

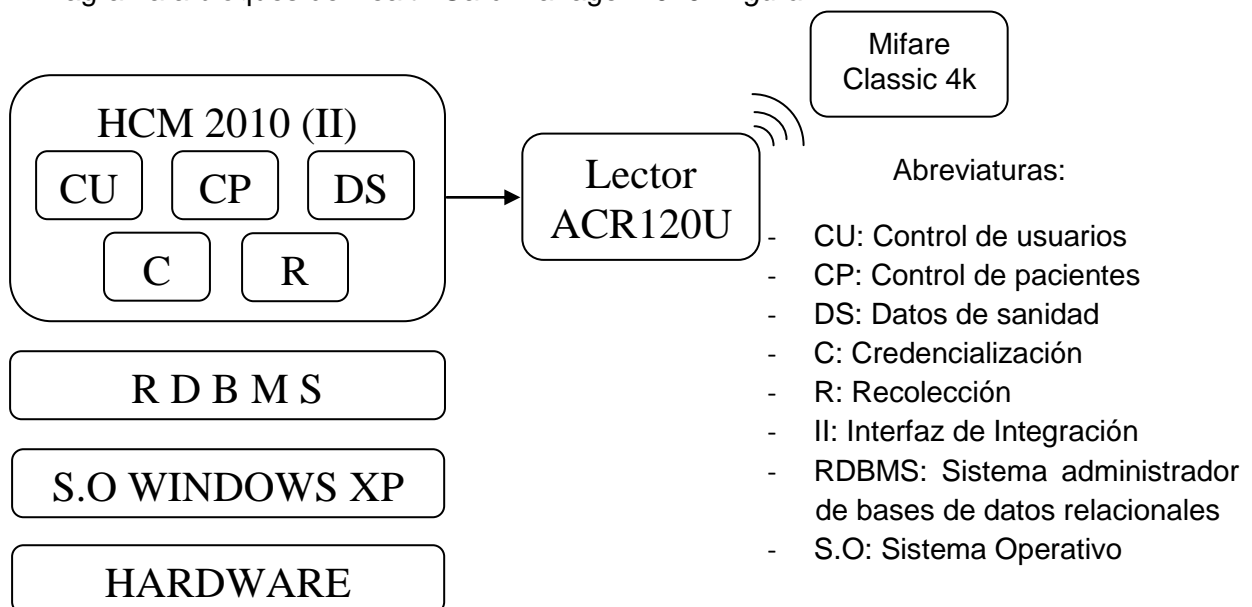


Figura 1. Bloques de la aplicación (Hardware - Software)



## ESPECIFICACIONES TÉCNICAS

Para la implementación de Health Card Manager 2010 se necesitara:

- Lector/Grabador de identificación por radiofrecuencia (ACR120U)
  - Interfaz USU Plug and Play
  - Fuente de alimentación desde USB
  - Funcionalidad de lectura y escritura
  - Distancia de lectura/escritura 5 cm
  - Funcionalidad anticolidión
  - Compatible con las tarjetas según la norma ISO 14443 tipo A y B



- Tarjetas Mifare Classic 4k
  - Distancia de operación: 10 cm (Depende de la antena)
  - Frecuencia de operación: 13.56 MHz
  - Transferencia de datos: 106 Kbit/s
  - 4 Kbyte, organizados en 32 sectores con 4 bloques y 8 sectores con 16 bloques (Cada bloque consta de 16 bytes)
  - Retención de datos: 10 años.
  - Ciclos de escritura: 100, 000
  - Numero de serie único para cada tarjeta



## ANÁLISIS Y DISEÑO DE LA BASE DE DATOS RELACIONAL

- Diccionario de datos

### USUARIOS

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDUSUARIO	Identificador de usuario	Identificador	
CEDULA	Cedula profesional si es medico	Número	
CURP	Clave única de registro de población	Texto	
NOMBRE	Nombre(s)	Texto	
APELLIDO_PATERNO	Apellido Paterno	Texto	
APELLIDO_MATERNO	Apellido Materno	Texto	
IDESPECIALIDAD	Identificador especialidad médica	Identificador	CAT_ESPECIALIDADES
IDSUB_ESPECIALIDAD	Identificador subespecialidad médica	Identificador	CAT_ESPECIALIDADES
IDCLUES	Identificador de CLUES	Identificador	CAT_CLUES
IDROL	Identificador de rol	Identificador	CAT_ROLES
USUARIO	Nombre de usuario	Texto	
CONTRASENA	Contraseña de usuario	Texto	
FECHA_ALTA	Fecha de alta	Fecha	
FECHA_BAJA	Fecha de baja	Fecha	
ACTIVADO	Usuario activado	Boléano	

### PACIENTES

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDPACIENTE	Identificador de paciente	Identificador	
NUMERO_PACIENTE	Numero de paciente	Número	
NOMBRE	Nombre(s)	Texto	
APELLIDO_PATERNO	Apellido Paterno	Texto	
APELLIDO_MATERNO	Apellido Materno	Texto	
CURP	Clave única de registro de población	Texto	
FECHA_NACIMIENTO	Fecha de nacimiento	Fecha	
IDNIVEL_SOCIOECONOMICO	Identificador de nivel socioeconómico	Identificador	CAT_NIVELES_SOCIOECONOMICOS
IDVIVIENDA	Identificador de vivienda	Identificador	CAT_VIVIENDAS
IDGRUPO_ETNICO	Identificador de grupo étnico	Identificador	CAT_GRUPOS_ETNICOS
IDRELIGION	Identificador de religión	Identificador	CAT_RELIGIONES
IDSEXO	Identificador de sexo	Identificador	CAT_SEXOS
IDSANGRE	Identificador de tipo de sangre	Identificador	CAT_SANGRE
IDDOMICILIO	Identificador de domicilio	Identificador	DOMICILIOS
FECHA_ALTA	Fecha de alta	Fecha	
FECHA_BAJA	Fecha de baja	Fecha	
TELEFONO_2	Teléfono particular	Texto	

## DOMICILIOS

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDDOMICILIO	Identificador de domicilio	Identificador	
IDTIPO_DOMICILIO	Identificador de tipo de domicilio	Identificador	CAT_TIPOS_DOMICILIO
CALLE	Calle	Texto	
NUMERO_EXT	Número exterior	Texto	
NUMERO_INT	Número interior	Texto	
COLONIA	Colonia	Texto	
IDMUNICIPIO_DELEGACION	Identificador de municipio/delegación	Identificador	CAT_MUNICIPIOS_DELEGACIONES
IDESTADO	Identificador de estado	Identificador	CAT_ESTADOS
CODIGO_POSTAL	Código postal	Número	
TELEFONO_1	Teléfono de casa	Texto	

## AUDITORIA

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDAUDITORIA	Identificador de auditoria	Identificador	
IDUSUARIO	Identificador de usuario	Identificador	USUARIOS
IDMODULO	Identificador de módulo	Identificador	CAT_MODULOS
IDMOVIMIENTO	Identificador de movimiento	Identificador	CAT_MOVIMIENTOS
DESCRIPCION	Descripción de auditoria	Texto	
FECHA	Fecha de auditoria	Fecha	
HORA	Hora de auditoria	Hora	

## CREDENCIALES

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDCREDENCIAL	Identificador de credencial	Identificador	
IDPACIENTE	Identificador de paciente	Identificador	PACIENTES
NUMERO_CREDENCIAL	Número de credencial	Texto	
FECHA_ALTA	Fecha de alta	Fecha	
FECHA_BAJA	Fecha de baja	Fecha	

## CAT\_ENFERMEDADES

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDENFERMEDAD	Identificador de enfermedad	Identificador	
IDGRUPO_ENFERMEDAD	Identificador grupo de enfermedad	Identificador	CAT_GRUPOS_ENFERMEDADES
ENFERMEDAD	Nombre de enfermedad	Texto	

## CAT\_DISCAPACIDADES

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDDISCAPACIDAD	Identificador de discapacidad	Identificador	
IDGRUPO_DISCAPACIDAD	Identificador grupo de discapacidad	Identificador	CAT_GRUPOS_DISCAPACIDADES
DISCAPACIDAD	Nombre de discapacidad	Texto	

**CAT\_MUNICIPIOS\_DELEGACIONES**

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDMUNICIPIO_DELEGACION	Identificador de municipio/delegación	Identificador	
IDESTADO	Identificador de estado	Identificador	CAT_ESTADOS
MUNICIPIO_DELEGACION	Nombre de municipio o delegación	Texto	

**CAT\_CLUES**

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDCLUES	Identificador de CLUES	Identificador	
CLUES	Abreviatura de CLUES	Texto	
DESCRIPCION	Clave única de establecimiento en salud	Texto	

**CAT\_GRUPOS\_ETNICOS**

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDGRUPO_ETNICO	Identificador de grupo étnico	Identificador	
GRUPO_ETNICO	Nombre de grupo étnico	Texto	

**CAT\_SANGRE**

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDSANGRE	Identificador de tipo de sangre	Identificador	
SANGRE	Nombre de tipo de sangre	Texto	

**CAT\_SEXOS**

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDSEXO	Identificador de sexo	Identificador	
SEXO	Nombre de sexo	Texto	

**CAT\_GRUPOS\_ENFERMEDADES**

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDGRUPO_ENFERMEDAD	Identificador de grupo de enfermedad	Identificador	
GRUPO_ENFERMEDAD	Nombre de grupo de enfermedad	Texto	

**CAT\_TIPOS\_DOMICILIO**

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDTIPO_DOMICILIO	Identificador de tipo de domicilio	Identificador	
TIPO_DOMICILIO	Nombre de tipo de domicilio	Texto	

**CAT\_ESTADOS**

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDESTADO	Identificador de estado	Identificador	
ESTADO	Nombre de estado	Texto	

### CAT\_NIVELES\_SOCIOECONOMICOS

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDNIVEL_SOCIOECONOMICO	Identificador de nivel socioeconómico	Identificador	
NIVEL_SOCIOECONOMICO	Nombre de tipo de domicilio	Texto	

### CAT\_GRUPOS\_DISCAPACIDADES

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDGRUPO_DISCAPACIDAD	Identificador de grupo de discapacidad	Identificador	
GRUPO_DISCAPACIDAD	Nombre de grupo de discapacidad	Texto	

### CAT\_RELIGIONES

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDRELIGION	Identificador de religión	Identificador	
RELIGION	Nombre de religión	Texto	

### CAT\_ESPECIALIDADES

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDGRUPO_ESPECIALIDAD	Identificador de grupo de especialidad	Identificador	
GRUPO_ESPECIALIDAD	Nombre de grupo de especialidad	Texto	

### CAT\_ROLES

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDROL	Identificador de rol	Identificador	
ROL	Nombre de rol	Texto	

### CAT\_MOVIMIENTOS

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDMOVIMIENTO	Identificador de movimiento	Identificador	
MOVIMIENTO	Nombre de movimiento	Texto	

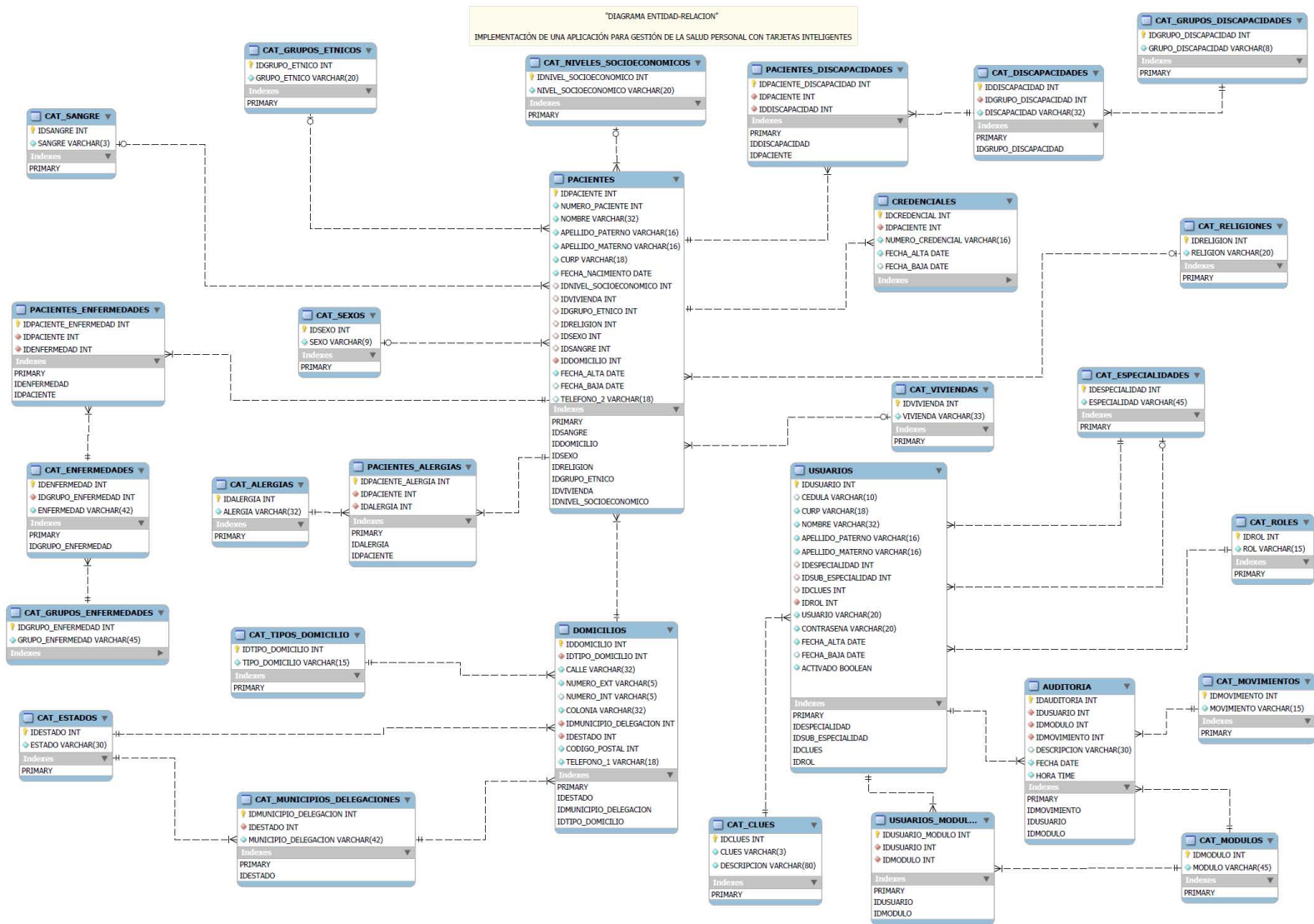
### CAT\_MODULOS

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDMODULO	Identificador de módulo	Identificador	
MODULO	Nombre de módulo	Texto	

### CAT\_ALERGIAS

Nombre del atributo	Descripción	Tipo de dato	Fuente
IDALERGIA	Identificador de alergia	Identificador	
ALERGIA	Nombre de alergia	Texto	

- Diagrama Entidad Relación



- Esquema Físico

```
CREATE DATABASE DBHEALTHCARD;
```

```
USE DBHEALTHCARD;
```

```
CREATE TABLE CAT_SANGRE(IDSANGRE INT NOT NULL AUTO_INCREMENT PRIMARY KEY,SANGRE VARCHAR(3) NOT NULL);
```

```
CREATE TABLE CAT_GRUPOS_ETNICOS(IDGRUPO_ETNICO INT NOT NULL AUTO_INCREMENT PRIMARY KEY,GRUPO_ETNICO VARCHAR(20) NOT NULL);
```

```
CREATE TABLE CAT_SEXOS(IDSEXO INT NOT NULL AUTO_INCREMENT PRIMARY KEY,SEXO VARCHAR(9) NOT NULL);
```

```
CREATE TABLE CAT_GRUPOS_ENFERMEDADES(IDGRUPO_ENFERMEDAD INT NOT NULL AUTO_INCREMENT PRIMARY KEY, GRUPO_ENFERMEDAD VARCHAR(45) NOT NULL);
```

```
CREATE TABLE CAT_ENFERMEDADES(IDENFERMEDAD INT NOT NULL AUTO_INCREMENT PRIMARY KEY, IDGRUPO_ENFERMEDAD INT NOT NULL, ENFERMEDAD VARCHAR(42) NOT NULL, FOREIGN KEY (IDGRUPO_ENFERMEDAD) REFERENCES CAT_GRUPOS_ENFERMEDADES (IDGRUPO_ENFERMEDAD));
```

```
CREATE TABLE CAT_ALERGIAS(IDALERGIA INT NOT NULL AUTO_INCREMENT PRIMARY KEY, ALERGIA VARCHAR(32) NOT NULL);
```

```
CREATE TABLE CAT_TIPOS_DOMICILIO(IDTIPO_DOMICILIO INT NOT NULL AUTO_INCREMENT PRIMARY KEY, TIPO_DOMICILIO VARCHAR(15) NOT NULL);
```

```
CREATE TABLE CAT_ESTADOS(IDESTADO INT NOT NULL AUTO_INCREMENT PRIMARY KEY, ESTADO VARCHAR(30) NOT NULL);
```

```
CREATE TABLE CAT_MUNICIPIOS_DELEGACIONES(IDMUNICIPIO_DELEGACION INT NOT NULL AUTO_INCREMENT PRIMARY KEY, IDESTADO INT NOT NULL, MUNICIPIO_DELEGACION VARCHAR(42) NOT NULL, FOREIGN KEY (IDESTADO) REFERENCES CAT_ESTADOS (IDESTADO));
```

```
CREATE TABLE CAT_NIVELES_SOCIOECONOMICOS(IDNIVEL_SOCIOECONOMICO INT NOT NULL AUTO_INCREMENT PRIMARY KEY,NIVEL_SOCIOECONOMICO VARCHAR(20));
```

```
CREATE TABLE CAT_GRUPOS_DISCAPACIDADES(IDGRUPO_DISCAPACIDAD INT NOT NULL AUTO_INCREMENT PRIMARY KEY, GRUPO_DISCAPACIDAD VARCHAR(8) NOT NULL);
```

```
CREATE TABLE CAT_DISCAPACIDADES(IDDISCAPACIDAD INT NOT NULL AUTO_INCREMENT PRIMARY KEY, IDGRUPO_DISCAPACIDAD INT NOT NULL, DISCAPACIDAD VARCHAR(32) NOT NULL, FOREIGN KEY(IDGRUPO_DISCAPACIDAD) REFERENCES CAT_GRUPOS_DISCAPACIDADES(IDGRUPO_DISCAPACIDAD));
```

```
CREATE TABLE CAT_RELIGIONES(IDRELIGION INT NOT NULL AUTO_INCREMENT PRIMARY KEY, RELIGION VARCHAR(20) NOT NULL);
```

```
CREATE TABLE CAT_ESPECIALIDADES(IDESPECIALIDAD INT NOT NULL AUTO_INCREMENT PRIMARY KEY, ESPECIALIDAD VARCHAR(45) NOT NULL);
```

```
CREATE TABLE CAT_VIVIENDAS(IDVIVIENDA INT NOT NULL AUTO_INCREMENT PRIMARY KEY, VIVIENDA VARCHAR(33));
```

```
CREATE TABLE CAT_ROLES(IDROL INT NOT NULL AUTO_INCREMENT PRIMARY KEY, ROL VARCHAR(15) NOT NULL);
```

```
CREATE TABLE CAT_CLUES(IDCLUES INT NOT NULL AUTO_INCREMENT PRIMARY KEY, CLUES VARCHAR(3) NOT NULL, DESCRIPCION VARCHAR(80));
```

```
CREATE TABLE CAT_MOVIMIENTOS(IDMOVIMIENTO INT NOT NULL AUTO_INCREMENT PRIMARY KEY, MOVIMIENTO VARCHAR(15) NOT NULL);
```

```
CREATE TABLE CAT_MODULOS(IDMODULO INT NOT NULL AUTO_INCREMENT PRIMARY KEY, MODULO VARCHAR(45) NOT NULL);
```

```
CREATE TABLE DOMICILIOS(IDDOMICILIO INT NOT NULL AUTO_INCREMENT PRIMARY KEY, IDTIPO_DOMICILIO INT NOT NULL, CALLE VARCHAR(32) NOT NULL, NUMERO_EXT VARCHAR(5) NOT NULL, NUMERO_INT VARCHAR(5), COLONIA VARCHAR(32) NOT NULL, IDMUNICIPIO_DELEGACION INT NOT NULL, IDESTADO INT NOT NULL, CODIGO_POSTAL INT NOT NULL, TELEFONO_1 VARCHAR(18) NOT NULL, FOREIGN KEY(IDTIPO_DOMICILIO) REFERENCES CAT_TIPOS_DOMICILIO(IDTIPO_DOMICILIO), FOREIGN KEY(IDMUNICIPIO_DELEGACION) REFERENCES
```



CAT\_MUNICIPIOS\_DELEGACIONES (IDMUNICIPIO\_DELEGACION), FOREIGN KEY (IDESTADO) REFERENCES CAT\_ESTADOS(IDESTADO));

CREATE TABLE PACIENTES(IDPACIENTE INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, NUMERO\_PACIENTE INT NOT NULL, NOMBRE VARCHAR(32) NOT NULL, APELLIDO\_PATERNO VARCHAR(16) NOT NULL, APELLIDO\_MATERNO VARCHAR(16) NOT NULL, CURP VARCHAR(18) NOT NULL, FECHA\_NACIMIENTO DATE NOT NULL, IDNIVEL\_SOCIOECONOMICO INT, IDVIVIENDA INT, IDGRUPO\_ETNICO INT, IDRELIGION INT, IDSEXO INT, IDSANGRE INT, IDDOMICILIO INT NOT NULL, FECHA\_ALTA DATE NOT NULL, FECHA\_BAJA DATE, TELEFONO\_2 VARCHAR(18), FOREIGN KEY (IDNIVEL\_SOCIOECONOMICO) REFERENCES CAT\_NIVELES\_SOCIOECONOMICOS (IDNIVEL\_SOCIOECONOMICO), FOREIGN KEY (IDVIVIENDA) REFERENCES CAT\_VIVIENDAS (IDVIVIENDA), FOREIGN KEY (IDGRUPO\_ETNICO) REFERENCES CAT\_GRUPOS\_ETNICOS (IDGRUPO\_ETNICO), FOREIGN KEY (IDRELIGION) REFERENCES CAT\_RELIGIONES (IDRELIGION), FOREIGN KEY (IDSEXO) REFERENCES CAT\_SEXOS (IDSEXO), FOREIGN KEY (IDDOMICILIO) REFERENCES DOMICILIOS (IDDOMICILIO), FOREIGN KEY (IDSANGRE) REFERENCES CAT\_SANGRE (IDSANGRE));

CREATE TABLE USUARIOS(IDUSUARIO INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, CEDULA VARCHAR(10), CURP VARCHAR(18) NOT NULL, NOMBRE VARCHAR(32) NOT NULL, APELLIDO\_PATERNO VARCHAR(16) NOT NULL, APELLIDO\_MATERNO VARCHAR(16) NOT NULL, IDESPECIALIDAD INT, IDSUB\_ESPECIALIDAD INT, IDCLUES INT, IDROL INT NOT NULL, USUARIO VARCHAR(20) NOT NULL, CONTRASENA VARCHAR(20) NOT NULL, FECHA\_ALTA DATE NOT NULL, FECHA\_BAJA DATE, ACTIVADO BOOL NOT NULL, FOREIGN KEY (IDESPECIALIDAD) REFERENCES CAT\_ESPECIALIDADES (IDESPECIALIDAD), FOREIGN KEY (IDSUB\_ESPECIALIDAD) REFERENCES CAT\_ESPECIALIDADES (IDESPECIALIDAD), FOREIGN KEY (IDCLUES) REFERENCES CAT\_CLUES (IDCLUES), FOREIGN KEY (IDROL) REFERENCES CAT\_ROLES (IDROL));

CREATE TABLE USUARIOS\_MODULOS(IDUSUARIO\_MODULO INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, IDUSUARIO INT NOT NULL, IDMODULO INT NOT NULL, FOREIGN KEY (IDUSUARIO) REFERENCES USUARIOS (IDUSUARIO), FOREIGN KEY (IDMODULO) REFERENCES CAT\_MODULOS (IDMODULO));

CREATE TABLE CREDENCIALES(IDCREDENCIAL INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, IDPACIENTE INT NOT NULL, NUMERO\_CREDENCIAL VARCHAR(16) NOT NULL, FECHA\_ALTA DATE NOT NULL, FECHA\_BAJA DATE, FOREIGN KEY (IDPACIENTE) REFERENCES PACIENTES (IDPACIENTE));

```
CREATE TABLE AUDITORIA(IDAUDITORIA INT NOT NULL AUTO_INCREMENT PRIMARY KEY, IDUSUARIO INT NOT NULL, IDMODULO INT NOT NULL, IDMOVIMIENTO INT NOT NULL, DESCRIPCION VARCHAR(30), FECHA DATE NOT NULL, HORA TIME NOT NULL, FOREIGN KEY(IDMOVIMIENTO) REFERENCES CAT_MOVIMIENTOS(IDMOVIMIENTO), FOREIGN KEY(IDUSUARIO) REFERENCES USUARIOS(IDUSUARIO), FOREIGN KEY (IDMODULO) REFERENCES CAT_MODULOS(IDMODULO));
```

```
CREATE TABLE PACIENTES_ALERGIAS(IDPACIENTE_ALERGIA INT NOT NULL AUTO_INCREMENT PRIMARY KEY, IDPACIENTE INT NOT NULL, IDALERGIA INT NOT NULL, FOREIGN KEY(IDPACIENTE) REFERENCES PACIENTES(IDPACIENTE), FOREIGN KEY(IDALERGIA) REFERENCES CAT_ALERGIAS(IDALERGIA));
```

```
CREATE TABLE PACIENTES_DISCAPACIDADES(IDPACIENTE_DISCAPACIDAD INT NOT NULL AUTO_INCREMENT PRIMARY KEY, IDPACIENTE INT NOT NULL, IDDISCAPACIDAD INT NOT NULL, FOREIGN KEY(IDPACIENTE) REFERENCES PACIENTES(IDPACIENTE), FOREIGN KEY(IDDISCAPACIDAD) REFERENCES CAT_DISCAPACIDADES (IDDISCAPACIDAD));
```

```
CREATE TABLE PACIENTES_ENFERMEDADES(IDPACIENTE_ENFERMEDAD INT NOT NULL AUTO_INCREMENT PRIMARY KEY, IDPACIENTE INT NOT NULL, IDENFERMEDAD INT NOT NULL, FOREIGN KEY(IDPACIENTE) REFERENCES PACIENTES(IDPACIENTE), FOREIGN KEY(IDENFERMEDAD) REFERENCES CAT_ENFERMEDADES (IDENFERMEDAD));
```

```
INSERT INTO CAT_GRUPOS_ETNICOS(GRUPO_ETNICO)VALUES ('NINGUNO'),('CHATINOS'), ('CHICHIMECOS'),('CHINANTECOS'),('CHOCHOPOPOLOCA'), ('CHONTALES'),('CHUJ'),('COCHIMÍ'),('CORAS'),('CUCAPÁ'),('CUICATECOS'),('GUAJIROS'), ('HUASTECOS'),('HUAVES'),('HUICHOLES'),('IXCATECOS'),('JALALTECOS'),('KIKAPÚS'), ('KILIWA'),('KUMIAI'),('MAYAS'),('MAYOS'),('MATLATZINCOS'),('MAZAHUAS'), ('MAZATECOS'), ('MIXES'),('MIXTECOS'),('MUZGOS'),('NAHUATLACAS'),('OCUILTECOS'), ('OTOMÍS'),('O´ODHAM DE CABORCA'),('PAIPAI'),('PAME'),('PÁPAGOS'),('PIMAS'), ('POPOLOCA'),('PURÉPECHAS'),('SERIS'),('TEPEHUANOS'),('TARAHUMARAS'), ('TEQUISTLATECOS'),('TLAPANECOS'),('TOTONACOS'),('TRIQUIS'),('YAQUIS'), ('ZAPOTECOS'),('ZOQUES');
```

```
INSERT INTO CAT_SANGRE(SANGRE) VALUES ('O-'),('O+'),('A-'),('A+'),('B-'),('B+'),('AB-'), ('AB+');
```

```
INSERT INTO CAT_SEXOS(SEXO) VALUES('MASCULINO'),('FEMENINO');
```

```
INSERT INTO CAT_TIPOS_DOMICILIO(TIPO_DOMICILIO) VALUES ('PRIMARIO'),
('TRABAJO'),('REFERENCIA');
```

```
INSERT INTO CAT_RELIGIONES(RELIGION) VALUES ('NINGUNA'),('ADVENTISTA'),
('BAUTISTA'),('BUDISTA'),('CATÓLICA'),('EVANGÉLICA'),('ISLÁMICA'),('JUDIA'), ('KRISHNA'),
('MENONITA'),('METODISTA'),('ORTODOXA'),('PENTECOSTAL'),('PRESBITERIANA'),
('TESTIGOS DE JEHOVÁ');
```

```
INSERT INTO CAT_ESTADOS(ESTADO)VALUES('AGUASCALIENTES'),('BAJA
CALIFORNIA'),('BAJA CALIFORNIA SUR'),('CAMPECHE'),('CHIAPAS'),('CHIHUAHUA'),
('COAHUILA'),('COLIMA'),('DISTRITO FEDERAL'), ('DURANGO'),('GUANAJUATO'),
('GUERRERO'), ('HIDALGO'),('JALISCO'), ('ESTADO DE MÉXICO'),('MICHOACÁN'),
('MORELOS'), ('NAYARIT'),('NUEVO LEÓN'),('OAXACA'),('PUEBLA'),('QUERÉTARO'),
('QUINTANA ROO'),('SAN LUIS POTOSÍ'), ('SINALOA'),('SONORA'),('TABASCO'),
('TAMAULIPAS'), ('TLAXCALA'),('VERACRUZ'),('YUCATÁN'),('ZACATECAS');
```

```
INSERT INTO CAT_MUNICIPIOS_DELEGACIONES(MUNICIPIO_DELEGACION,IDESTADO)
VALUES('AGUASCALIENTES',1),('ASIENTOS CALVILLO',1), ('COSÍO',1),('EL LLANO',1),
('JESÚS MARÍA',1),('PABELLÓN DE ARTEAGA',1),('RINCÓN DE ROMOS',1),
('SAN FRANCISCO DE LOS ROMO',1),('SAN JOSÉ DE GRACIA',1),('TEPEZALÁ',1),
('ENSENADA',2),('MEXICALI',2),('PLAYAS DE ROSARITO',2),('TECATE',2),('TIJUANA',2),
('COMONDÚ',3),('LA PAZ',3),('LORETO',3),('LOS CABOS',3),('MULEGÉ',3), ('CALAKMUL',4),
('CALKINÍ',4),('CAMPECHE',4),('CANDELARIA',4),('CARMEN',4), ('CHAMPOTÓN',4),
('ESCÁRCEGA',4),('HECELCHAKÁN',4),('HOPELCHÉN',4),('PALIZADA',4),('TENABO',4),
('ACACOYAGUA',5),('ACALA',5),('ACAPETAHUA',5),('ALDAMA',5),('ALTAMIRANO',5),
('AMATÁN',5),('AMATENANGO DE LA FRONTERA',5),('AMATENANGO DEL
VALLE',5),('ANGEL ALBINO CORZO',5),('ARRIAGA',5),('BEJUCAL DE OCAMPO',5),
('BELLA VISTA',5),('BENEMÉRITO DE LAS AMÉRICAS',5),('BERRIOZÁBAL',5),('BOCHIL',5),
('CACAHOATÁN',5),('CATAZAJÁ',5),('CHALCHIHUITÁN',5),('CHAMULA',5),('CHANAL',5),
('CHAPULTENANGO',5),('CHENALHÓ',5),('CHIAPA DE CORZO',5),('CHIAPILLA',5),
('CHICOASÉN',5),('CHICOMUSELO',5),('CHILÓN',5),('CINTALAPA',5),('COAPILLA',5),
('COMITÁN DE DOMÍNGUEZ',5), ('COPAINALÁ',5),('EL BOSQUE',5),('EL PORVENIR',5),
('ESCUINTLA',5),('FRANCISCO LEÓN',5),('FRONTERA COMALAPA',5),
('FRONTERA HIDALGO',5),('HUEHUETÁN',5),('HUITIUPÁN',5),('HUIXTÁN',5),('HUIXTLA',5),
('IXHUATÁN',5),('IXTACOMITÁN',5),('IXTAPA',5),('IXTAPANGAJOYA',5),('JIQUIPILAS',5),
('JITOTOL',5),('JUÁREZ',5),('LA CONCORDIA',5),('LA GRANDEZA',5),
('LA INDEPENDENCIA',5),('LA LIBERTAD',5),('LA TRINITARIA',5),('LARRÁINZAR',5),
```

('LAS MARGARITAS',5),('LAS ROSAS',5),('MAPASTEPEC',5),('MARAVILLA TENEJAPA',5),  
('MARQUÉS DE COMILLAS',5),('MAZAPA DE MADERO',5),('MAZATÁN',5),('METAPA',5),  
('MITONTIC',5),('MONTECRISTO DE GUERRERO',5),('MOTOZINTLA',5),('NICOLÁS RUÍZ',5),  
('OCOSINGO',5),('OCOTEPEC',5),('OCOZOCOAUTLA DE ESPINOSA',5),  
('OSTUACÁN OSUMACINTA',5),('OXCHUC',5),('PALENQUE',5),('PANTELHÓ',5),  
('PANTEPEC',5),('PICHUCALCO',5),('PIJIAPAN',5),  
('PUEBLO NUEVO SOLISTAHUACÁN',5),('RAYÓN',5),('REFORMA',5),('SABANILLA',5),  
('SALTO DE AGUA',5),('SAN ANDRÉS DURAZNAL',5),('SAN CRISTÓBAL DE LAS CASAS',5),  
('SAN FERNANDO',5),('SAN JUAN CANCUC',5),('SAN LUCAS',5),('SANTIAGO EL PINAR',5),  
('SILTEPEC SIMOJOVEL',5),('SITALÁ',5),('SOCOLTENANGO',5),('SOLOSUCHIAPA',5),  
('SOYALÓ',5),('SUCHIAPA',5),('SUCHIATE',5),('SUNUAPA',5),('TAPACHULA',5),  
('TAPALAPA',5),('TAPILULA',5),('TECPATÁN',5),('TENEJAPA',5),('TEOPISCA',5),('TILA',5),  
('TONALÁ',5),('TOTOLAPA',5),('TUMBALÁ',5),('TUXTLA CHICO',5),('TUXTLA  
GUTIÉRREZ',5),('TUZANTÁN',5),('TZIMOL',5),('UNIÓN JUÁREZ',5),  
('VENUSTIANO CARRANZA',5),('VILLA COMALTITLÁN',5),('VILLA CORZO',5),  
('VILLAFLORES',5),('YAJALÓN',5),('ZINACANTÁN',5),('AHUMADA',6),('ALDAMA',6),  
('ALLENDE',6),('AQUILES SERDÁN',6),('ASCENSIÓN',6),('BACHÍNIVA',6),('BALLEZA',6),  
('BATOPILAS',6),('BOCOYNA',6),('BUENAVENTURA',6),('CAMARGO',6),('CARICHÍ',6),  
('CASAS GRANDES',6),('CHÍNIPAS',6),('CHIHUAHUA',6),('CORONADO',6),  
('COYAME DEL SOTOL',6),('CUAUHTÉMOC',6),('CUSIHUIRIACHI',6),('DELICIAS',6),('DR.  
BELISARIO DOMÍNGUEZ',6),('EL TULE',6),('GALEANA',6),('GÓMEZ FARÍAS',6),  
('GRAN MORELOS',6),('GUACHOCHI',6),('GUADALUPE',6),('GUADALUPE Y CALVO',6),  
('GUAZAPARES',6),('GUERRERO',6),('HIDALGO DEL PARRAL',6),('HUEJOTITÁN',6),  
('IGNACIO ZARAGOZA',6),('JANOS',6),('JIMÉNEZ',6),('JUÁREZ',6),('JULIMES',6),  
('LA CRUZ',6),('LÓPEZ',6),('MADERA',6),('MAGUARICHI',6),('MANUEL BENAVIDES',6),  
('MATACHÍ',6),('MATAMOROS',6),('MEOQUI',6),('MORELOS',6),('MORIS',6),('NAMQUIPA',6),  
('NONOAVA',6),('NUEVO CASAS GRANDES',6),('OCAMPO',6),('OJINAGA',6),  
('PRAXEDIS G. GUERRERO',6),('RIVA PALACIO',6),('ROSALES',6),  
('ROSARIO',6),('SAN FRANCISCO DE BORJA',6),('SAN FRANCISCO DE CONCHOS',6),  
('SAN FRANCISCO DEL ORO',6),('SANTA BÁRBARA',6),('SANTA ISABEL',6),('SATEVÓ',6),  
('SAUCILLO',6),('TEMÓSACHI',6),('URIQUE',6),('URUACHI',6),('VALLE DE ZARAGOZA',6),  
('ABASOLO',7),('ACUÑA',7),('ALLENDE',7),('ARTEAGA',7),('CANDELA',7),  
('CASTAÑOS',7),('CUATRO CIÉNEGAS',7),('ESCOBEDO',7),('FRANCISCO I. MADERO',7),  
('FRONTERA',7),('GENERAL CEPEDA',7),('GUERRERO',7),('HIDALGO',7),('JIMÉNEZ',7),  
('JUÁREZ',7),('LAMADRID',7),('MATAMOROS',7),('MÚZQUIZ',7),('MONCLOVA',7),  
('MORELOS',7),('NADADORES',7),('NAVA',7),('OCAMPO',7),('PARRAS',7),  
('PIEDRAS NEGRAS',7),('PROGRESO',7),('RAMOSARIZPE',7),('SABINAS',7),  
('SACRAMENTO',7),('SALTILLO',7),('SAN BUENAVENTURA',7),('SAN JUAN DE SABINAS',7),  
('SAN PEDRO',7),('SIERRA MOJADA',7),('TORREÓN',7),('VIESCA',7),('VILLA UNIÓN',7),  
('ZARAGOZA',7),('ARMERÍA',8),('COLIMA',8),('COMALA',8),('COQUIMATLÁN',8),

('CUAUHTÉMOC',8),('IXTLAHUACÁN',8),('MANZANILLO',8),('MINATITLÁN',8),('TECOMÁN',8),  
('VILLA DE ÁLVAREZ',8),('AZCAPOTZALCO',9),('ÁLVARO OBREGÓN',9),  
('BENITO JUÁREZ',9),('COYOACÁN',9),('CUAJIMALPA DE MORELOS',9),  
('CUAUHTÉMOC',9),('GUSTAVO A. MADERO',9),('IZTACALCO',9),('IZTAPALAPA',9),  
('LA MAGDALENA CONTRERAS',9),('MIGUEL HIDALGO',9),('MILPA ALTA',9),('TLALPAN',9),  
('TLÁHUAC',9),('VENUSTIANO CARRANZA',9),('XOCHIMILCO',9),('CANATLÁN',10),  
('CANELAS',10),('CONETO DE COMONFORT',10),('CUENCAMÉ',10),('DURANGO',10),  
('EL ORO',10),('GÓMEZ PALACIO',10),('GENERAL SIMÓN BOLÍVAR',10),  
('GUADALUPE VICTORIA',10),('GUANACEVÍ',10),('HIDALGO',10),('INDÉ',10),('LERDO',10),  
('MAPIMÍ',10),('MEZQUITAL',10),('NAZAS',10),('NOMBRE DE DIOS',10),('NUEVO IDEAL',10),  
('OCAMPO',10),('OTÁEZ',10),('PÁNUCO DE CORONADO',10),('PEÑÓN BLANCO',10),  
('POANAS',10),('PUEBLO NUEVO',10),('RODEO',10),  
('SAN BERNARDO',10),('SAN DIMAS',10),('SAN JUAN DE GUADALUPE',10),  
('SAN JUAN DEL RÍO',10),('SAN LUIS DEL CORDERO',10),  
('SAN PEDRO DEL GALLO',10),('SANTA CLARA',10),('SANTIAGO PAPASQUIARO',10),  
('SÚCHIL',10),('TAMAZULA',10),('TEPEHUANES',10),('TLAHUALILO',10),('TOPIA',10),  
('VICENTE GUERRERO',10),('ABASOLO',11),('ACÁMBARO',11),('APASEO EL ALTO',11),  
('APASEO EL GRANDE',11),('ATARJEJA',11),  
('CELAYA',11),('COMONFORT',11),('CORONEO',11),('CORTAZAR',11),('CUERÁMARO',11),  
('DOCTOR MORA',11),('DOLORES HIDALGO',11),('GUANAJUATO',11),('HUANÍMARO',11),  
('IRAPUATO',11),('JARAL DEL PROGRESO',11),('JERÉCUARO',11),('LEÓN',11),  
('MANUEL DOBLADO',11),('MOROLEÓN',11),('OCAMPO',11),('PÉNJAMO',11),  
('PUEBLO NUEVO',11),('PURÍSIMA DEL RINCÓN',11),('ROMITA',11),('SALAMANCA',11),  
('SALVATIERRA',11),('SAN DIEGO DE LA UNIÓN',11),('SAN FELIPE',11),  
('SAN FRANCISCO DEL RINCÓN',11),('SAN JOSÉ ITURBIDE',11),  
('SAN LUIS DE LA PAZ',11),('SAN MIGUEL DE ALLENDE',11),('SANTA CATARINA',11),  
('SANTA CRUZ DE JUVENTINO ROSAS',11),('SANTIAGO MARAVATÍO',11),('SILAO',11),  
('TARANDACUAO',11),('TARIMORO',11),('TIERRA BLANCA',11),  
('URIANGATO',11),('VALLE DE SANTIAGO',11),('VICTORIA',11),  
('VILLAGRÁN',11),('XICHÚ',11),('YURIRIA',11),('ACAPULCO DE JUÁREZ',12),  
('ACATEPEC',12),('AHUACUOTZINGO',12),('AJUCHITLÁN DEL PROGRESO',12),  
('ALCOZAUCA DE GUERRERO',12),('ALPOYECA',12),('APAXTLA',12),('ARCELIA',12),  
('ATENANGO DEL RÍO',12),('ATLAMAJALCINGO DEL MONTE',12),('ATLIXTAC',12),  
('ATOYAC DE ÁLVAREZ',12),('AYUTLA DE LOS LIBRES',12),('AZOYÚ',12),  
('BENITO JUÁREZ',12),('BUENAVISTA DE CUÉLLAR',12),('CHILAPA DE ÁLVAREZ',12),  
('CHILPANCINGO DE LOS BRAVO',12),('COAHUAYUTLA DE JOSÉ MARÍA IZAZAGA',12),  
('COCHOAPA EL GRANDE',12),('COCULA',12),('COPALA',12),('COPALILLO',12),  
('COPANATOYAC',12),('COYUCA DE BENÍTEZ',12),('COYUCA DE CATALÁN',12),  
('CUAJINICUILAPA',12),('CUALÁC',12),('CUAUTEPEC',12),

('CUETZALA DEL PROGRESO',12), ('CUTZAMALA DE PINZÓN',12),('EDUARDO NERI',12),  
( 'FLORENCIO VILLARREAL',12),('GENERAL CANUTO A. NERI',12),  
( 'GENERAL HELIODORO CASTILLO',12),('HUAMUXTITLÁN',12),  
( 'HUITZUCO DE LOS FIGUEROA',12),('IGUALA DE LA INDEPENDENCIA',12),  
( 'IGUALAPA',12),('ILIATENCO',12),('IXCATEOPAN DE CUAUHTÉMOC',12),  
( 'JOSÉ AZUETA',12), ('JOSÉ JOAQUÍN DE HERRERA',12),('JUAN R. ESCUDERO',12),  
( 'JUCHITÁN',12),('LA UNIÓN DE ISIDORO MONTES DE OCA',12),  
( 'LEONARDO BRAVO',12),('MALINALTEPEC',12),('MARQUELIA',12),  
( 'MÁRTIR DE CUILAPAN',12), ('MÁRTIR DE CUILAPAN',12), ('MOCHITLÁN',12),  
( 'OLINALÁ',12),('OMETEPEC',12),('PEDRO ASCENCIO ALQUISIRAS',12),('PETATLÁN',12),  
( 'PILCAYA',12),('PUNGARABATO',12),('QUECHULTENANGO',12),('SAN LUIS ACATLÁN',12),  
( 'SAN MARCOS',12),('SAN MIGUEL TOTOLAPAN',12),('TAXCO DE ALARCÓN',12),  
( 'TÉCPAN DE GALEANA',12),('TECOANAPA',12),('TEOLOAPAN',12),  
( 'TEPECOACUILCO DE TRUJANO',12),('TETIPAC',12),('TIXTLA DE GUERRERO',12),  
( 'TLACOACHISTLAHUACA',12),('TLACOAPA',12),('TLALCHAPA',12),  
( 'TLALIXTAQUILLA DE MALDONADO',12),('TLAPA DE COMONFORT',12),  
( 'TLAPEHUALA',12),('XALPATLÁHUAC',12),('XOCHIHUEHUETLÁN',12),  
( 'XOCHISTLAHUACA',12), ('ZAPOTITLÁN TABLAS',12),('ZIRÁNDARO',12),('ZITLALA',12),  
( 'ACATLÁN',13),('ACAXOCHITLÁN',13),('ACTOPAN',13),('AGUA BLANCA DE ITURBIDE',13),  
( 'AJACUBA',13),('ALFAJAYUCAN',13),('ALMOLOYA',13),('APAN',13),('ATITALAQUIA',13),  
( 'ATLAPEXCO',13),('ATOTONILCO DE TULA',13),('ATOTONILCO EL GRANDE',13),  
( 'CALNALI',13), ('CARDONAL',13),('CHAPANTONGO',13),('CHAPULHUACÁN',13),  
( 'CHILCUAUTLA',13),('CUAUTEPEC DE HINOJOSA',13),('EL ARENAL',13),  
( 'ELOXOCHITLÁN',13),('EMILIANO ZAPATA',13),('EPAZOYUCAN',13),  
( 'FRANCISCO I. MADERO',13), ('HUASCA DE OCAMPO',13),('HUAUTLA',13),  
( 'HUAZALINGO',13),('HUEHUETLA',13),('HUEJUTLA DE REYES',13),('HUICHAPAN',13),  
( 'IXMIQUILPAN',13),('JACALA DE LEDEZMA',13),('JALTOCÁN',13),('JUÁREZ HIDALGO',13),  
( 'LA MISIÓN',13),('LOLOTLA',13),('METEPEC',13),('METZTITLÁN',13),  
( 'MINERAL DE LA REFORMA',13),('MINERAL DEL CHICO',13),('MINERAL DEL MONTE',13),  
( 'MIXQUIAHUALA DE JUÁREZ',13),('MOLANGO DE ESCAMILLA',13),  
( 'NICOLÁS FLORES',13), ('NOPALA DE VILLAGRÁN',13),('OMITLÁN DE JUÁREZ',13),  
( 'PACHUCA DE SOTO',13),('PACULA',13),('PISAFLORES',13),  
( 'PROGRESO DE OBREGÓN',13),('SAN AGUSTÍN METZQUITITLÁN',13),  
( 'SAN AGUSTÍN TLAXIACA',13),('SAN BARTOLO TUTOTEPEC',13),  
( 'SAN FELIPE ORIZATLÁN',13), ('SAN SALVADOR',13),('SANTIAGO DE ANAYA',13),  
( 'SANTIAGO TULANTEPEC DE LUGO GUERRERO',13),('SINGUILUCAN',13),  
( 'TASQUILLO',13),('TECOZAUTLA',13),('TENANGO DE DORIA',13),('TEPEAPULCO',13),  
( 'TEPEHUACÁN DE GUERRERO',13),('TEPEJI DEL RÍO DE OCAMPO',13),

('TEPETILÁN',13),('TETEPANGO',13),('TEZONTEPEC DE ALDAMA',13),  
('TIANGUISTENGO',13),('TIZAYUCA',13),('TLAHUELILPAN',13), ('TLAHUILTEPA',13),  
('TLANALAPA',13),('TLANCHINOL',13),('TLAXCOAPAN',13),('TOLCAYUCA',13),  
('TULA DE ALLENDE',13),('TULANCINGO DE BRAVO',13),('VILLA DE TEZONTEPEC',13),  
('XOCHIATIPAN',13),('XOCHICOATLÁN',13),('YAHUALICA',13),  
('ZACUALTIPÁN DE ÁNGELES',13),('ZAPOTLÁN DE JUÁREZ',13),('ZEMPOALA',13),  
('ZIMAPÁN',13),('ACATIC',14),('ACATLÁN DE JUÁREZ',14),  
('AHUALULCO DE MERCADO',14),('AMACUECA',14),('AMATITÁN',14),('AMECA',14),  
('ARANDAS',14),('ATEMAJAC DE BRIZUELA',14),('ATENGO',14),('ATENGUILLO',14),  
('ATOTONILCO EL ALTO',14),('ATOYAC',14),('AUTLÁN DE NAVARRO',14),('AYOTLÁN',14),  
('AYUTLA',14),('BOLAÑOS',14),('CAÑADAS DE OBREGÓN',14),('CABO CORRIENTES',14),  
('CASIMIRO CASTILLO',14), ('CHAPALA',14),('CHIMALTITÁN',14),('CHIQUILISTLÁN',14),  
('CIHUATLÁN',14),('COCULA',14),('COLOTLÁN',14),  
('CONCEPCIÓN DE BUENOS AIRES',14),('CUAUTITLÁN DE GARCÍA BARRAGÁN',14),  
('CUAUTLA',14),('CUQUÍO',14),('DEGOLLADO',14),  
('EJUTLA',14),('EL ARENAL',14),('EL GRULLO',14),('EL LIMÓN',14),('EL SALTO',14),  
('ENCARNACIÓN DE DÍAZ',14),('ETZATLÁN',14),('GÓMEZ FARÍAS',14),  
('GUACHINANGO',14),('GUADALAJARA',14),  
('HOSTOTIPAQUILLO',14),('HUEJÚCAR',14),('HUEJUQUILLA EL ALTO',14),  
('IXTLAHUACÁN DE LOS MEMBRILLOS',14),('IXTLAHUACÁN DEL RÍO',14),  
('JALOSTOTITLÁN',14),('JAMAY',14),('JESÚS MARÍA',14),  
('JILOTLÁN DE LOS DOLORES',14),('JOCOTEPEC',14),('JUANACATLÁN',14),  
('JUCHITLÁN',14),('LA BARCA',14),('LA HUERTA',14),('LA MANZANILLA DE LA PAZ',14),  
('LAGOS DE MORENO',14),('MAGDALENA',14),('MASCOTA',14),('MAZAMITLA',14),  
('MEXTICACÁN',14),('MEZQUITIC',14),('MIXTLÁN',14),('OCOTLÁN',14),  
('OJUELOS DE JALISCO',14), ('PIHUAMO',14),('PONCITLÁN',14),('PUERTO VALLARTA',14),  
('QUITUPAN',14),('SAN CRISTÓBAL DE LA BARRANCA',14),  
('SAN DIEGO DE ALEJANDRÍA',14),('SAN GABRIEL',14),  
('SAN JUAN DE LOS LAGOS',14),('SAN JUANITO DE ESCOBEDO',14),  
('SAN JULIÁN',14),('SAN MARCOS',14),('SAN MARTÍN DE BOLAÑOS',14),  
('SAN MARTÍN HIDALGO',14),('SAN MIGUEL EL ALTO',14),  
('SAN SEBASTIÁN DEL OESTE',14),('SANTA MARÍA DE LOS ÁNGELES',14),  
('SANTA MARÍA DEL ORO',14),('SAYULA',14),('TALA',14),  
('TALPA DE ALLENDE',14),('TAMAZULA DE GORDIANO',14),('TAPALPA',14),  
('TECALITLÁN',14),('TECHALUTA DE MONTENEGRO',14),  
('TECOLOTLÁN',14),('TENAMAXTLÁN',14),('TEOCALTICHE',14),  
('TEOCUITATLÁN DE CORONA',14),('TEPATITLÁN DE MORELOS',14),  
('TEQUILA',14),('TEUCHITLÁN',14),('TIZAPÁN EL ALTO',14),  
('TLAJOMULCO DE ZÚÑIGA',14),('TLAQUEPAQUE',14),('TOLIMÁN',14),  
('TOMATLÁN',14),('TONALÁ',14),('TONAYA',14),('TONILA',14),('TOTATICHE',14),

('TOTOTLÁN',14),('TUXCACUESCO',14),('TUXCUECA',14),('TUXPAN',14),  
('UNIÓN DE SAN ANTONIO',14),('UNIÓN DE TULA',14),  
('VALLE DE GUADALUPE',14),('VALLE DE JUÁREZ',14),('VILLA CORONA',14),  
('VILLA GUERRERO',14),('VILLA HIDALGO',14),('VILLA PURIFICACIÓN',14),  
('YAHUALICA DE GONZÁLEZ GALLO',14),('ZACOALCO DE TORRES',14),('ZAPOPAN',14),  
('ZAPOTILCO',14),('ZAPOTITLÁN DE VADILLO',14),('ZAPOTLANEJO',14),  
('ZAPOTLÁN DEL REY',14),('ZAPOTLÁN EL GRANDE',14),  
('ACAMBAY',15),('ACOLMAN',15),('ACULCO',15),('ALMOLOYA DE ALQUISIRAS',15),  
('ALMOLOYA DE JUÁREZ',15),('ALMOLOYA DEL RÍO',15),  
('AMANALCO',15),('AMATEPEC',15),('AMECAMECA',15),('APAXCO',15),  
('ATENCO',15),('ATIZAPÁN',15),('ATIZAPÁN DE ZARAGOZA',15),  
('ATLACOMULCO',15),('ATLAUTLA',15),('AXAPUSCO',15),('AYAPANGO',15),('CALIMAYA',15),  
('CAPULHUAC',15),('CHALCO',15),('CHAPA DE MOTA',15),('CHAPULTEPEC',15),  
('CHIAUTLA',15),('CHICOLOAPAN',15),('CHICONCUAC',15),  
('CHIMALHUACÁN',15),('COACALCO DE BERRIOZÁBAL',15),('COATEPEC HARINAS',15),  
('COCOTITLÁN',15),('COYOTEPEC',15),('CUAUTITLÁN',15),('CUAUTITLÁN IZCALLI',15),  
('DONATO GUERRA',15),('ECATEPEC DE MORELOS',15),('ECATZINGO',15),  
('EL ORO',15),('HUEHUETOCA',15),('HUEYPOXTLA',15),('HUIXQUILUCAN',15),  
('ISIDRO FABELA',15),('ISIDRO FABELA',15),('IXTAPAN DE LA SAL',15),  
('IXTAPAN DEL ORO',15),('IXTLAHUACA',15),('JALTENCO',15),  
('JILOTEPEC',15),('JILOTZINGO',15),('JIQUIPILCO',15),('JOCOTITLÁN',15),  
('JOQUICINGO',15),('JUCHITEPEC',15),('LA PAZ',15),('LERMA',15),('LUVIANOS',15),  
('MALINALCO',15),('MELCHOR OCAMPO',15),('METEPEC',15),('MEXICALTZINGO',15),  
('MORELOS',15),('NAUCALPAN DE JUÁREZ',15),  
('NEXTLALPAN',15),('NEZAHUALCÓYOTL',15),('NICOLÁS ROMERO',15),  
('NOPALTEPEC',15),('OCOYOACAC',15),('OCUILAN',15),  
('OTUMBA',15),('OTZOLOAPAN',15),('OTZOLOTEPEC',15),('OZUMBA',15),('PAPALOTLA',15),  
('POLOTITLÁN',15),('RAYÓN',15),('SAN ANTONIO LA ISLA',15),  
('SAN FELIPE DEL PROGRESO',15),('SAN JOSÉ DEL RINCÓN',15),  
('SAN MARTÍN DE LAS PIRÁMIDES',15),('SAN MATEO ATENCO',15),  
('SAN SIMÓN DE GUERRERO',15),('SANTO TOMÁS',15),  
('SOYANIQUELIPAN DE JUÁREZ',15),('SULTEPEC',15),('TECÁMAC',15),('TEJUPILCO',15),  
('TEMAMATLA',15),('TEMASCALAPA',15),('TEMASCALCINGO',15),('TEMASCALTEPEC',15),  
('TEMOAYA',15),('TENANCINGO',15),('TENANGO DEL AIRE',15),  
('TENANGO DEL VALLE',15),('TEOLOYUCÁN',15),('TEOTIHUACÁN',15),  
('TEPETLAOXTOC',15),('TEPETLIXPA',15),('TEPOTZOTLÁN',15),('TEQUIXQUIAC',15),  
('TEXCALTITLÁN',15),('TEXCALYACAC',15),('TEXCOCO',15),  
('TEZOYUCA',15),('TIANGUISTENCO',15),('TIMILPAN',15),('TLALMANALCO',15),  
('TLALNEPANTLA DE BAZ',15),('TLATLAYA',15),('TOLUCA',15),('TONANITLA',15),  
('TONATICO',15),('TULTEPEC',15),('TULTITLÁN',15),('VALLE DE BRAVO',15),



('VALLE DE CHALCO SOLIDARIDAD',15),('VILLA DE ALLENDE',15),  
('VILLA DEL CARBÓN',15),('VILLA GUERRERO',15),('VILLA VICTORIA',15),  
('XALATLACO',15),('XONACATLÁN',15),('ZACAZONAPAN',15),  
('ZACUALPAN',15),('ZINACANTEPEC',15),('ZUMPAHUACÁN',15),('ZUMPANGO',15),  
('ACUITZIO',16),('AGUILILLA',16),('ANGAMACUTIRO',16),('ANGANGUEO',16),  
('APATZINGÁN',16),('APORO',16),('AQUILA',16),('ARIO',16),('ARTEAGA',16),  
('ÁLVARO OBREGÓN',16),('BRISEÑAS',16),('BUENAVISTA',16),('CARÁCUARO',16),  
('CHARAPAN',16),('CHARO',16),('CHAVINDA',16),('CHERÁN',16),('CHILCHOTA',16),  
('CHINICUILA',16),('CHUCÁNDIRO',16),('CHURINTZIO',16),('CHURUMUCO',16),  
('COAHUAYANA',16),('COALCOMÁN DE VÁZQUEZ PALLARES',16),('COENEO',16),  
('COJUMATLÁN DE RÉGULES',16),('CONTEPEC',16),('COPÁNDARO',16),  
('COTIJA',16),('CUITZEO',16),('ECUANDUREO',16),('EPITACIO HUERTA',16),  
('ERONGARÍCUARO',16),('GABRIEL ZAMORA',16),('HIDALGO',16),  
('HUANDACAREO',16),('HUANIQUEO',16),('HUETAMO',16),('HUIRAMBA',16),  
('INDAPARAPEO',16),('IRIMBO',16),('IXTLÁN',16),('JACONA',16),('JIMÉNEZ',16),  
('JIQUILPAN',16),('JOSÉ SIXTO VERDUZCO',16),('JUÁREZ',16),('JUNGAPEO',16),  
('LA HUACANA',16),('LA PIEDAD',16),  
('LAGUNILLAS',16),('LÁZARO CÁRDENAS',16),('LOS REYES',16),('MADERO',16),  
('MARAVATÍO',16),('MARCOS CASTELLANOS',16),('MÚGICA',16),('MORELIA',16),  
('MORELOS',16),('NAHUATZEN',16),('NOCUPÉTARO',16),  
('NUEVO PARANGARICUTIRO',16),('NUEVO URECHO',16),('NUMARÍAN',16),('OCAMPO',16),  
('PAJACUARÁN',16),('PANINDÍCUARO',16),('PARACHO',16),('PARÁCUARO',16),  
('PÁTZCUARO',16),('PENJAMILLO',16),('PERIBÁN',16),('PURÉPERO',16),  
('PURUÁNDIRO',16),('QUERÉNDARO',16),('QUIROGA',16),('SAHUAYO',16),  
('SALVADOR ESCALANTE',16),('SAN LUCAS',16),('SANTA ANA MAYA',16),  
('SENGUIO',16),('SUSUPUATO',16),('TACÁMBARO',16),('TANCÍTARO',16),  
('TANGAMANDAPIO',16),('TANGANCÍCUARO',16),('TANHUATO',16),('TARÍMBARO',16),  
('TARETAN',16),('TEPALCATEPEC',16),('TINGAMBATO',16),('TINGÜINDÍN',16),  
('TIQUICHEO DE NICOLÁS ROMERO',16),('TLALPUJAHUA',16),('TLAZAZALCA',16),  
('TOCUMBO',16),('TUMBISCATÍO',16),('TURICATO',16),('TUXPAN',16),('TUZANTLA',16),  
('TZINTZUNTZAN',16),('TZITZIO',16),('URUAPAN',16),('VENUSTIANO CARRANZA',16),  
('VILLAMAR',16),('VISTA HERMOSA',16),('YURÉCUARO',16),('ZACAPU',16),('ZAMORA',16),  
('ZINAPÉCUARO',16),('ZINÁPARO',16),('ZIRACUARETIRO',16),('ZITÁCUARO',16),  
('AMACUZAC',17),('ATLATLAHUCAN',17),('AXOCHIAPAN',17),('AYALA',17),  
('COATLÁN DEL RÍO',17),('CUAUTLA',17),('CUERNAVACA',17),('EMILIANO ZAPATA',17),  
('HUITZILAC',17),('JANTETELCO',17),('JIUTEPEC',17),('JOJUTLA',17),('JONACATEPEC',17),  
('MAZATEPEC',17),('MIACATLÁN',17),('OCUITUCO',17),('PUENTE DE IXTLA',17),  
('TEMIXCO',17),('TEMOAC',17),('TEPALcingo',17),('TEPOZTLÁN',17),('TETECALA',17),  
('TETELA DEL VOLCÁN',17),('TLALNEPANTLA',17),('TLALTIZAPÁN',17),  
('TLAQUILTENANGO',17),('TLAYACAPAN',17),('TOTOLAPAN',17),('XOCHITEPEC',17),

('YAUTEPEC',17),('YECAPIXTLA',17),('ZACATEPEC DE HIDALGO',17),  
('ZACUALPAN DE AMILPAS',17),('ACAPONETA',18),('AHUACATLÁN',18),  
('AMATLÁN DE CAÑAS',18),('BAHÍA DE BANDERAS',18),('COMPOSTELA',18),  
('DEL NAYAR',18),('HUAJICORI',18),('IXTLÁN DEL RÍO',18),('JALA',18),('LA YESCA',18),  
('ROSAMORADA',18),('RUÍZ',18),('SAN BLAS',18),('SAN PEDRO LAGUNILLAS',18),  
('SANTA MARÍA DEL ORO',18),('SANTIAGO IXCUINTLA',18),('TECUALA',18),('TEPIC',18),  
('TUXPAN',18),('XALISCO',18),('ABASOLO',19),('AGUALEGUAS',19),('ALLENDE',19),  
('ANÁHUAC',19),('APODACA',19),('ARAMBERRI',19),('BUSTAMANTE',19),  
('CADEREYTA JIMÉNEZ',19),('CARMEN',19),('CERRALVO',19),  
('CHINA',19),('CIÉNEGA DE FLORES',19),('DR. ARROYO',19),('DR. COSS',19),  
('DR. GONZÁLEZ',19),('GALEANA',19),('GARCÍA',19),('GRAL. BRAVO',19),  
('GRAL. ESCOBEDO',19),('GRAL. TERÁN',19),('GRAL. TREVIÑO',19),  
('GRAL. ZARAGOZA',19),('GRAL. ZUAZUA',19),('GUADALUPE',19),('HIDALGO',19),  
('HIGUERAS',19),('HUALAHUISES',19),('ITURBIDE',19),('JUÁREZ',19),  
('LAMPAZOS DE NARANJO',19),('LINARES',19),('LOS ALDAMAS',19),  
('LOS HERRERAS',19),('LOS RAMONES',19),('MARÍAN',19),('MELCHOR OCAMPO',19),  
('MIER Y NORIEGA',19),('MINA',19),('MONTEMORELOS',19),('MONTERREY',19),  
('PARÁS',19),('PESQUERÍA',19),('RAYONES',19),('SABINAS HIDALGO',19),  
('SALINAS VICTORIA',19),('SAN NICOLÁS DE LOS GARZA',19),  
('SAN PEDRO GARZA GARCÍA',19),('SANTA CATARINA',19),('SANTIAGO',19),  
('VALLECILLO',19),('VILLALDAMA',19),('ABEJONES',20),  
('ACATLÁN DE PÉREZ FIGUEROA',20),('ASUNCIÓN CACALOTEPEC',20),  
('ASUNCIÓN CUYOTEPEJI',20),('ASUNCIÓN IXTALTEPEC',20),  
('ASUNCIÓN NOCHIXTLÁN',20),('ASUNCIÓN OCOTLÁN',20),  
('ASUNCIÓN TLACOLULITA',20),('AYOQUEZCO DE ALDAMA',20),('AYOTZINTEPEC',20),  
('ÁNIMAS TRUJANO',20),('CALIHUALÁ',20),('CANDELARIA LOXICHA',20),  
('CAPULÁLPAM DE MÉNDEZ',20),('CHAHUITES',20),  
('CHALCATONGO DE HIDALGO',20),('CHIQUIHUITLÁN DE BENITO JUÁREZ',20),  
('CIÉNEGA DE ZIMATLÁN',20),('CIUDAD IXTEPEC',20),('COATECAS ALTAS',20),  
('COICOYÁN DE LAS FLORES',20),('CONCEPCIÓN BUENAVISTA',20),  
('CONCEPCIÓN PÁPALO',20),('CONSTANCIA DEL ROSARIO',20),('COSOLAPA',20),  
('COSOLTEPEC',20),('CUILÁPAM DE GUERRERO',20),  
('CUYAMECALCO VILLA DE ZARAGOZA',20),('EL BARRIO DE LA SOLEDAD',20),  
('EL ESPINAL',20),('ELOXOCHITLÁN DE FLORES MAGÓN',20),  
('FRESNILLO DE TRUJANO',20),('GUADALUPE DE RAMÍREZ',20),  
('GUADALUPE ETLA',20),('GUELATAO DE JUÁREZ',20),('GUEVEA DE HUMBOLDT',20),  
('HEROICA CIUDAD DE EJUTLA DE CRESPO',20),  
('HEROICA CIUDAD DE HUAJUAPAN DE LEÓN',20),('HEROICA CIUDAD DE TLAXIACO',20),  
('HUAUTEPEC',20),('HUAUTLA DE JIMÉNEZ',20),('IXPANTEPEC NIEVES',20),  
('IXTLÁN DE JUÁREZ',20),('JUCHITÁN DE ZARAGOZA',20),('LA COMPAÑÍA',20),

('LA PE',20),('LA REFORMA',20),('LA TRINIDAD VISTA HERMOSA',20),('LOMA BONITA',20),  
('MAGDALENA APASCO',20),('MAGDALENA JALTEPEC',20),('MAGDALENA MIXTEPEC',20),  
('MAGDALENA OCOTLÁN',20),('MAGDALENA PEÑASCO',20),('MAGDALENA TEITIPAC',20),  
('MAGDALENA TEQUISISTLÁN',20),('MAGDALENA TLACOTEPEC',20),  
('MAGDALENA YODOCONO DE PORFIRIO DÍAZ',20),('MAGDALENA ZAHUATLÁN',20),  
('MARISCALA DE JUÁREZ',20),('MATÍAS ROMERO AVENDAÑO',20),  
('MAZATLÁN VILLA DE FLORES',20),('MÁRTIRES DE TACUBAYA',20),  
('MESONES HIDALGO',20),('MIAHUATLÁN DE PORFIRIO DÍAZ',20),  
('MIXISTLÁN DE LA REFORMA',20),('MONJAS',20),('NATIVIDAD',20),  
('NAZARENO ETLA',20),('NEJAPA DE MADERO',20),  
('NUEVO ZOQUIÁPAM',20),('OAXACA DE JUÁREZ',20),('OCOTLÁN DE MORELOS',20),  
('PINOTEPA DE DON LUIS',20),('PLUMA HIDALGO',20),  
('PUTLA VILLA DE GUERRERO',20),('REFORMA DE PINEDA',20),('REYES ETLA',20),  
('ROJAS DE CUAUHTÉMOC',20),('SALINA CRUZ',20),  
('SAN AGUSTÍN AMATENGO',20),('SAN AGUSTÍN ATENANGO',20),  
('SAN AGUSTÍN CHAYUCO',20),('SAN AGUSTÍN DE LAS JUNTAS',20),  
('SAN AGUSTÍN ETLA',20),('SAN AGUSTÍN LOXICHA',20),  
('SAN AGUSTÍN TLACOTEPEC',20),('SAN AGUSTÍN YATARENI',20),  
('SAN ANDRÉS CABECERA NUEVA',20),('SAN ANDRÉS DINICUITI',20),  
('SAN ANDRÉS HUAXPALTEPEC',20),('SAN ANDRÉS HUAYÁPAM',20),  
('SAN ANDRÉS IXTLAHUACA',20),('SAN ANDRÉS LAGUNAS',20),  
('SAN ANDRÉS NUXIÑO',20),('SAN ANDRÉS PAXTLÁN',20),('SAN ANDRÉS SINAXTLA',20),  
('SAN ANDRÉS SOLAGA',20),('SAN ANDRÉS TEOTILÁLPAM',20),  
('SAN ANDRÉS TEPETLAPA',20),('SAN ANDRÉS YAÁ',20),('SAN ANDRÉS ZABACHE',20),  
('SAN ANDRÉS ZAUTLA',20),('SAN ANTONINO CASTILLO VELASCO',20),  
('SAN ANTONINO EL ALTO',20),('SAN ANTONINO MONTE VERDE',20),  
('SAN ANTONIO ACUTLA',20),('SAN ANTONIO DE LA CAL',20),  
('SAN ANTONIO HUITEPEC',20),('SAN ANTONIO NANAHUATÍPAM',20),  
('SAN ANTONIO SINICAHUA',20),('SAN ANTONIO TEPETLAPA',20),  
('SAN BALTAZAR CHICHICÁPAM',20),('SAN BALTAZAR LOXICHA',20),  
('SAN BALTAZAR YATZACHI EL BAJO',20),('SAN BARTOLO COYOTEPEC',20),  
('SAN BARTOLO SOYALTEPEC',20),('SAN BARTOLO YAUTEPEC',20),  
('SAN BARTOLOMÉ AYAUTLA',20),('SAN BARTOLOMÉ LOXICHA',20),  
('SAN BARTOLOMÉ QUIALANA',20),('SAN BARTOLOMÉ YUCUAÑE',20),  
('SAN BARTOLOMÉ ZOOGOCHO',20),('SAN BERNARDO MIXTEPEC',20),  
('SAN BLAS ATEMPA',20),('SAN CARLOS YAUTEPEC',20),  
('SAN CRISTÓBAL AMATLÁN',20),('SAN CRISTÓBAL AMOLTEPEC',20),  
('SAN CRISTÓBAL LACHIRIOAG',20),('SAN CRISTÓBAL SUCHIXTLAHUACA',20),  
('SAN DIONISIO DEL MAR',20),('SAN DIONISIO OCOTEPEC',20),  
('SAN DIONISIO OCOTLÁN',20),('SAN ESTEBAN ATATLAHUCA',20),

('SAN FELIPE JALAPA DE DÍAZ',20),('SAN FELIPE TEJALÁPAM',20),  
('SAN FELIPE USILA',20),('SAN FRANCISCO CAHUACUÁ',20),  
('SAN FRANCISCO CAJONOS',20),('SAN FRANCISCO CHAPULAPA',20),  
('SAN FRANCISCO CHINDÚA',20),('SAN FRANCISCO DEL MAR',20),  
('SAN FRANCISCO HUEHUETLÁN',20),('SAN FRANCISCO IXHUATÁN',20),  
('SAN FRANCISCO JALTEPETONGO',20),('SAN FRANCISCO LACHIGOLÓ',20),  
('SAN FRANCISCO LOGUECHE',20),('SAN FRANCISCO NUXAÑO',20),  
('SAN FRANCISCO OZOLOTEPEC',20),('SAN FRANCISCO SOLA',20),  
('SAN FRANCISCO TELIXTLAHUACA',20),('SAN FRANCISCO TEOPAN',20),  
('SAN FRANCISCO TLAPANCINGO',20),('SAN GABRIEL MIXTEPEC',20),  
('SAN ILDEFONSO AMATLÁN',20),('SAN ILDEFONSO SOLA',20),  
('SAN ILDEFONSO VILLA ALTA',20),('SAN JACINTO AMILPAS',20),  
('SAN JACINTO TLACOTEPEC',20),('SAN JERÓNIMO COATLÁN',20),  
('SAN JERÓNIMO SILACAYOAPILLA',20),('SAN JERÓNIMO SOSOLA',20),  
('SAN JERÓNIMO TAVICHE',20),('SAN JERÓNIMO TECÓATL',20),  
('SAN JERÓNIMO TLACOCHAHUAYA',20),('SAN JORGE NUCHITA',20),  
('SAN JOSÉ AYUQUILA',20),('SAN JOSÉ CHILTEPEC',20),('SAN JOSÉ DEL PEÑASCO',20),  
('SAN JOSÉ DEL PROGRESO',20),('SAN JOSÉ ESTANCIA GRANDE',20),  
('SAN JOSÉ INDEPENDENCIA',20),('SAN JOSÉ LACHIGUIRI',20),  
('SAN JOSÉ TENANGO',20),('SAN JUAN ACHIUTLA',20),('SAN JUAN ATEPEC',20),  
('SAN JUAN ÑUMÍ',20),('SAN JUAN BAUTISTA ATATLAHUCA',20),  
('SAN JUAN BAUTISTA COIXTLAHUACA',20),('SAN JUAN BAUTISTA CUICATLÁN',20),  
('SAN JUAN BAUTISTA GUELACHE',20),('SAN JUAN BAUTISTA JAYACATLÁN',20),  
('SAN JUAN BAUTISTA LO DE SOTO',20),('SAN JUAN BAUTISTA SUCHITEPEC',20),  
('SAN JUAN BAUTISTA TLACHICHILCO',20),  
('SAN JUAN BAUTISTA TLACOATZINTEPEC',20),('SAN JUAN BAUTISTA TUXTEPEC',20),  
('SAN JUAN BAUTISTA VALLE NACIONAL',20),('SAN JUAN CACAHUATEPEC',20),  
('SAN JUAN CHICOMEZÚCHIL',20),('SAN JUAN CHILATECA',20),  
('SAN JUAN CIENEGUILLA',20),('SAN JUAN COATZÓSPAM',20),  
('SAN JUAN COLORADO',20),('SAN JUAN COMALTEPEC',20),('SAN JUAN COTZOCÓN',20),  
('SAN JUAN DE LOS CUÉS',20),('SAN JUAN DEL ESTADO',20),('SAN JUAN DEL RÍO',20),  
('SAN JUAN DIUXÍ',20),('SAN JUAN EVANGELISTA ANALCO',20),  
('SAN JUAN GUELAVÍA',20),('SAN JUAN GUICHICOVI',20),  
('SAN JUAN IHUALTEPEC',20),('SAN JUAN JUQUILA MIXES',20),  
('SAN JUAN JUQUILA VIJANOS',20),('SAN JUAN LACHAO',20),  
('SAN JUAN LACHIGALLA',20),('SAN JUAN LAJARCIA',20),('SAN JUAN LALANA',20),  
('SAN JUAN MAZATLÁN',20),('SAN JUAN MIXTEPEC - DTO. 08 -',20),  
('SAN JUAN MIXTEPEC - DTO. 26 -',20),('SAN JUAN OZOLOTEPEC',20),  
('SAN JUAN PETLAPA',20),('SAN JUAN QUIAHIJE',20),('SAN JUAN QUIOTEPEC',20),  
('SAN JUAN SAYULTEPEC',20),('SAN JUAN TABAÁ',20),('SAN JUAN TAMAZOLA',20),

('SAN JUAN TEITA',20),('SAN JUAN TEITIPAC',20),('SAN JUAN TEPEUXILA',20),  
('SAN JUAN TEPOSCOLULA',20),('SAN JUAN YAEÉ',20),  
('SAN JUAN YATZONA',20),('SAN JUAN YUCUITA',20),('SAN LORENZO',20),  
('SAN LORENZO ALBARRADAS',20),('SAN LORENZO CACAOTEPEC',20),  
('SAN LORENZO CUAUNECUILTITLA',20),('SAN LORENZO TEXMELÚCAN',20),  
('SAN LORENZO VICTORIA',20),('SAN LUCAS CAMOTLÁN',20),('SAN LUCAS OJITLÁN',20),  
('SAN LUCAS QUIAVINÍ',20),('SAN LUCAS ZOQUIÁPAM',20),  
('SAN LUIS AMATLÁN',20),('SAN MARCIAL OZOLOTEPEC',20),  
('SAN MARCOS ARTEAGA',20),('SAN MARTÍN DE LOS CANSECOS',20),  
('SAN MARTÍN HUAMELÚPAM',20),('SAN MARTÍN ITUNYOSO',20),  
('SAN MARTÍN LACHILÁ',20),('SAN MARTÍN PERAS',20),  
('SAN MARTÍN TILCAJETE',20),('SAN MARTÍN TOXPALAN',20),  
('SAN MARTÍN ZACATEPEC',20),('SAN MATEO CAJONOS',20),('SAN MATEO DEL MAR',20),  
('SAN MATEO ETLATONGO',20),('SAN MATEO NEJÁPAM',20),('SAN MATEO PEÑASCO',20),  
('SAN MATEO PIÑAS',20),('SAN MATEO RÍO HONDO',20),  
('SAN MATEO SINDIHUI',20),('SAN MATEO TLAPILTEPEC',20),  
('SAN MATEO YOLOXOCHITLÁN',20), ('SAN MELCHOR BETAZA',20),  
('SAN MIGUEL ACHIUTLA',20),('SAN MIGUEL AHUEHUETITLÁN',20),  
('SAN MIGUEL ALOÁPAM',20),('SAN MIGUEL AMATITLÁN',20),  
('SAN MIGUEL AMATLÁN',20),('SAN MIGUEL CHICAHUA',20),  
('SAN MIGUEL CHIMALAPA',20), ('SAN MIGUEL COATLÁN',20),  
('SAN MIGUEL DEL PUERTO',20),('SAN MIGUEL DEL RÍO',20),('SAN MIGUEL EJUTLA',20),  
('SAN MIGUEL EL GRANDE',20),('SAN MIGUEL HUAUTLA',20),  
('SAN MIGUEL MIXTEPEC',20),('SAN MIGUEL PANIXTLAHUACA',20),  
('SAN MIGUEL PERAS',20),('SAN MIGUEL PIEDRAS',20),  
('SAN MIGUEL QUETZALTEPEC',20),('SAN MIGUEL SANTA FLOR',20),  
('SAN MIGUEL SOYALTEPEC',20),('SAN MIGUEL SUCHIXTEPEC',20),  
('SAN MIGUEL TECOMATLÁN',20),('SAN MIGUEL TENANGO',20),  
('SAN MIGUEL TEQUIXTEPEC',20),('SAN MIGUEL TILQUIÁPAM',20),  
('SAN MIGUEL TLACAMAMA',20),('SAN MIGUEL TLACOTEPEC',20),  
('SAN MIGUEL TULANCINGO',20),('SAN MIGUEL YOTAO',20),('SAN NICOLÁS',20),  
('SAN NICOLÁS HIDALGO',20),('SAN PABLO COATLÁN',20),  
('SAN PABLO CUATRO VENADOS',20),('SAN PABLO ETLA',20),('SAN PABLO HUITZO',20),  
('SAN PABLO HUIXTEPEC',20),('SAN PABLO MACUILTIANGUIS',20),  
('SAN PABLO TIJALTEPEC',20),('SAN PABLO VILLA DE MITLA',20),  
('SAN PABLO YAGANIZA',20),('SAN PEDRO AMUZGOS',20),('SAN PEDRO APÓSTOL',20),  
('SAN PEDRO ATOYAC',20),('SAN PEDRO CAJONOS',20),  
('SAN PEDRO COMITANCILLO',20),('SAN PEDRO COXCALTEPEC CÁNTAROS',20),  
('SAN PEDRO EL ALTO',20),('SAN PEDRO HUAMELULA',20),  
('SAN PEDRO HUILOTEPEC',20),('SAN PEDRO IXCATLÁN',20),

('SAN PEDRO IXTLAHUACA',20),('SAN PEDRO JALTEPETONGO',20),  
('SAN PEDRO JICAYÁN',20),('SAN PEDRO JOCOTIPAC',20),  
('SAN PEDRO JUCHATENGO',20),('SAN PEDRO MÁRTIR',20),  
('SAN PEDRO MÁRTIR QUIECHAPA',20),('SAN PEDRO MÁRTIR YUCUXACO',20),  
('SAN PEDRO MIXTEPEC - DTO. 22 -',20),('SAN PEDRO MIXTEPEC - DTO. 26 -',20),  
('SAN PEDRO MOLINOS',20),('SAN PEDRO NOPALA',20),  
('SAN PEDRO OCOPETATILLO',20),('SAN PEDRO OCOTEPEC',20),  
('SAN PEDRO POCHUTLA',20),('SAN PEDRO QUIATONI',20),  
('SAN PEDRO SOCHIÁPAM',20),('SAN PEDRO TAPANATEPEC',20),  
('SAN PEDRO TAVICHE',20),('SAN PEDRO TEOZACOALCO',20),  
('SAN PEDRO TEUTILA',20),('SAN PEDRO TIDAÁ',20),('SAN PEDRO TOPILTEPEC',20),  
('SAN PEDRO TOTOLAPA',20),('SAN PEDRO Y SAN PABLO AYUTLA',20),  
('SAN PEDRO Y SAN PABLO TEPOSCOLULA',20),  
('SAN PEDRO Y SAN PABLO TEQUIXTEPEC',20),('SAN PEDRO YANERI',20),  
('SAN PEDRO YÓLOX',20),('SAN PEDRO YUCUNAMA',20),  
('SAN RAYMUNDO JALPAN',20),('SAN SEBASTIÁN ABASOLO',20),  
('SAN SEBASTIÁN COATLÁN',20),('SAN SEBASTIÁN IXCAPA',20),  
('SAN SEBASTIÁN NICANANDUTA',20),('SAN SEBASTIÁN RÍO HONDO',20),  
('SAN SEBASTIÁN TECOMAXTLAHUACA',20),('SAN SEBASTIÁN TEITIPAC',20),  
('SAN SEBASTIÁN TUTLA',20),('SAN SIMÓN ALMOLONGAS',20),  
('SAN SIMÓN ZAHUATLÁN',20),('SAN VICENTE COATLÁN',20),  
('SAN VICENTE LACHIXÍO',20),('SAN VICENTE NUÑÚ',20),('SANTA ANA',20),  
('SANTA ANA ATEIXTLAHUACA',20),('SANTA ANA CUAUHTÉMOC',20),  
('SANTA ANA DEL VALLE',20),('SANTA ANA TAVELA',20),('SANTA ANA TLAPACOYAN',20),  
('SANTA ANA YARENI',20),('SANTA ANA ZEGACHE',20),('SANTA CATALINA QUIERÍ',20),  
('SANTA CATARINA CUIXTLA',20),('SANTA CATARINA IXTEPEJI',20),  
('SANTA CATARINA JUQUILA',20),('SANTA CATARINA LACHATAO',20),  
('SANTA CATARINA LOXICHA',20),('SANTA CATARINA MECHOACÁN',20),  
('SANTA CATARINA MINAS',20),('SANTA CATARINA QUIANÉ',20),  
('SANTA CATARINA QUIOQUITANI',20),('SANTA CATARINA TAYATA',20),  
('SANTA CATARINA TICUÁ',20),('SANTA CATARINA YOSONOTÚ',20),  
('SANTA CATARINA ZAPOQUILA',20),('SANTA CRUZ ACATEPEC',20),  
('SANTA CRUZ AMILPAS',20),('SANTA CRUZ DE BRAVO',20),  
('SANTA CRUZ ITUNDUJIA',20),('SANTA CRUZ MIXTEPEC',20),  
('SANTA CRUZ NUNDACO',20),('SANTA CRUZ PAPALUTLA',20),  
('SANTA CRUZ TACACHE DE MINA',20),('SANTA CRUZ TACAHUA',20),  
('SANTA CRUZ TAYATA',20),('SANTA CRUZ XITLA',20),('SANTA CRUZ XOXOCOTLÁN',20),  
('SANTA CRUZ ZENZONTEPEC',20),('SANTA GERTRUDIS',20),  
('SANTA INÉS DE ZARAGOZA',20),('SANTA INÉS DEL MONTE',20),  
('SANTA INÉS YATZECHE',20),('SANTA LUCÍA DEL CAMINO',20),

('SANTA LUCÍA MIAHUATLÁN',20),('SANTA LUCÍA MONTEVERDE',20),  
('SANTA LUCÍA OCOTLÁN',20),('SANTA MAGDALENA JICOTLÁN',20),  
('SANTA MARÍA ALOTEPEC',20),('SANTA MARÍA APAZCO',20),  
('SANTA MARÍA ATZOMPA',20),('SANTA MARÍA CAMOTLÁN',20),  
('SANTA MARÍA CHACHOÁPAM',20),('SANTA MARÍA CHILCHOTLA',20),  
('SANTA MARÍA CHIMALAPA',20),('SANTA MARÍA COLOTEPEC',20),  
('SANTA MARÍA CORTIJO',20),('SANTA MARÍA COYOTEPEC',20),  
('SANTA MARÍA DEL ROSARIO',20),('SANTA MARÍA DEL TULE',20),  
('SANTA MARÍA ECATEPEC',20),('SANTA MARÍA GUELACÉ',20),  
('SANTA MARÍA GUIENAGATI',20),('SANTA MARÍA HUATULCO',20),  
('SANTA MARÍA HUAZOLOTITLÁN',20),('SANTA MARÍA IPALAPA',20),  
('SANTA MARÍA IXCATLÁN',20),('SANTA MARÍA JACATEPEC',20),  
('SANTA MARÍA JALAPA DEL MARQUÉS',20),('SANTA MARÍA JALTIANGUIS',20),  
('SANTA MARÍA LA ASUNCIÓN',20),('SANTA MARÍA LACHIXÍO',20),  
('SANTA MARÍA MIXTEQUILLA',20),('SANTA MARÍA NATIVITAS',20),  
('SANTA MARÍA NDUAYACO',20),('SANTA MARÍA OZOLOTEPEC',20),  
('SANTA MARÍA PÁPALO',20),('SANTA MARÍA PENOLES',20),  
('SANTA MARÍA PETAPA',20),('SANTA MARÍA QUIEGOLANI',20),('SANTA MARÍA SOLA',20),  
('SANTA MARÍA TATALTEPEC',20),('SANTA MARÍA TECOMAVACA',20),  
('SANTA MARÍA TEMAXCALAPA',20),('SANTA MARÍA TEMAXCALTEPEC',20),  
('SANTA MARÍA TEOPOXCO',20),('SANTA MARÍA TEPANTLALI',20),  
('SANTA MARÍA TEXCATITLÁN',20),('SANTA MARÍA TLAHUITOLTEPEC',20),  
('SANTA MARÍA TLALIXTAC',20),('SANTA MARÍA TONAMECA',20),  
('SANTA MARÍA TOTOLAPILLA',20),('SANTA MARÍA XADANI',20),  
('SANTA MARÍA YALINA',20),('SANTA MARÍA YAVESÍA',20),  
('SANTA MARÍA YOLOTEPEC',20),('SANTA MARÍA YOSOYÚA',20),  
('SANTA MARÍA YUCUHITI',20),('SANTA MARÍA ZACATEPEC',20),  
('SANTA MARÍA ZANIZA',20),('SANTA MARÍA ZOQUITLÁN',20),  
('SANTIAGO AMOLTEPEC',20),('SANTIAGO APÓSTOL',20),('SANTIAGO APOALA',20),  
('SANTIAGO ASTATA',20),('SANTIAGO ATITLÁN',20),('SANTIAGO AYUQUILILLA',20),  
('SANTIAGO CACALOXTEPEC',20),('SANTIAGO CAMOTLÁN',20),  
('SANTIAGO CHAZUMBA',20),('SANTIAGO CHOÁPAM',20),  
('SANTIAGO COMALTEPEC',20),('SANTIAGO DEL RÍO',20),  
('SANTIAGO HUAJOLOTITLÁN',20),('SANTIAGO HUAUCLILLA',20),  
('SANTIAGO IHUITLÁN PLUMAS',20),('SANTIAGO IXCUINTEPEC',20),  
('SANTIAGO IXTAYUTLA',20),('SANTIAGO JAMILTEPEC',20),('SANTIAGO JOCOTEPEC',20),  
('SANTIAGO JUXTLAHUACA',20),('SANTIAGO LACHIGUIRI',20),('SANTIAGO LALOPA',20),  
('SANTIAGO LAOLLAGA',20),('SANTIAGO LAXOPA',20),('SANTIAGO LLANO GRANDE',20),  
('SANTIAGO MATATLÁN',20),('SANTIAGO MILTEPEC',20),('SANTIAGO MINAS',20),  
('SANTIAGO NACALTEPEC',20),('SANTIAGO NEJAPILLA',20),

('SANTIAGO NILTEPEC',20),('SANTIAGO NUNDICHE',20),('SANTIAGO NUYOÓ',20),  
('SANTIAGO PINOTEPA NACIONAL',20),('SANTIAGO SUCHILQUITONGO',20),  
('SANTIAGO TAMAZOLA',20),('SANTIAGO TAPEXTLA',20),('SANTIAGO TENANGO',20),  
('SANTIAGO TEPETLAPA',20),('SANTIAGO TETEPEC',20),  
('SANTIAGO TEXCALCINGO',20),('SANTIAGO TEXTITLÁN',20),  
('SANTIAGO TILANTONGO',20),('SANTIAGO TILLO',20),('SANTIAGO TLAZOYALTEPEC',20),  
('SANTIAGO XANICA',20),('SANTIAGO XIACUÍ',20),('SANTIAGO YAITEPEC',20),  
('SANTIAGO YAVEO',20),('SANTIAGO YOLOMÉCATL',20),  
('SANTIAGO YOSONDÚA',20),('SANTIAGO YUCUYACHI',20),  
('SANTIAGO ZACATEPEC',20),('SANTIAGO ZOOCHILA',20),  
('SANTO DOMINGO ALBARRADAS',20),('SANTO DOMINGO ARMENTA',20),  
('SANTO DOMINGO CHIHUITÁN',20),('SANTO DOMINGO DE MORELOS',20),  
('SANTO DOMINGO INGENIO',20),('SANTO DOMINGO IXCATLÁN',20),  
('SANTO DOMINGO NUXAÁ',20),('SANTO DOMINGO OZOLOTEPEC',20),  
('SANTO DOMINGO PETAPA',20),('SANTO DOMINGO ROAYAGA',20),  
('SANTO DOMINGO TEHUANTEPEC',20),('SANTO DOMINGO TEOJOMULCO',20),  
('SANTO DOMINGO TEPUXTEPEC',20),('SANTO DOMINGO TLATAYÁPAM',20),  
('SANTO DOMINGO TOMALTEPEC',20),('SANTO DOMINGO TONALÁ',20),  
('SANTO DOMINGO TONALTEPEC',20),('SANTO DOMINGO XAGACÍA',20),  
('SANTO DOMINGO YANHUITLÁN',20),('SANTO DOMINGO YODOHINO',20),  
('SANTO DOMINGO ZANATEPEC',20),('SANTO TOMÁS JALIEZA',20),  
('SANTO TOMÁS MAZALTEPEC',20),('SANTO TOMÁS OCOTEPEC',20),  
('SANTO TOMÁS TAMAZULAPAN',20),('SANTOS REYES NOPALA',20),  
('SANTOS REYES PÁPALO',20),('SANTOS REYES TEPEJILLO',20),  
('SANTOS REYES YUCUNÁ',20),('SILACAYOÁPAM',20),('SITIO DE XITLAPEHUA',20),  
('SOLEDAD ETLA',20),('TAMAZULÁPAM DEL ESPÍRITU SANTO',20),  
('TANETZE DE ZARAGOZA',20),('TANICHE',20),('TATALTEPEC DE VALDÉS',20),  
('TEOCOCUILCO DE MARCOS PÉREZ',20),('TEOTITLÁN DE FLORES MAGÓN',20),  
('TEOTITLÁN DEL VALLE',20),('TEOTONGO',20),('TEPELMEME VILLA DE MORELOS',20),  
('TEZOATLÁN DE SEGURA Y LUNA',20),('TLACOLULA DE MATAMOROS',20),  
('TLACOTEPEC PLUMAS',20),('TLALIXTAC DE CABRERA',20),  
('TOTONTEPEC VILLA DE MORELOS',20),('TRINIDAD ZAACHILA',20),  
('UNIÓN HIDALGO',20),('VALERIO TRUJANO',20),('VILLA DÍAZ ORDAZ',20),  
('VILLA DE CHILAPA DE DÍAZ',20),('VILLA DE ETLA',20),  
('VILLA DE TAMAZULÁPAM DEL PROGRESO',20),  
('VILLA DE TUTUTEPEC DE MELCHOR OCAMPO',20),('VILLA DE ZAACHILA',20),  
('VILLA HIDALGO',20),('VILLA SOLA DE VEGA',20),('VILLA TALEA DE CASTRO',20),  
('VILLA TEJÚPAM DE LA UNIÓN',20),('YAXE',20),('YOGANA',20),('VILLA DE ETLA',20),  
('VILLA TALEA DE CASTRO',20),('VILLA TEJÚPAM DE LA UNIÓN',20),('YAXE',20),



('YOGANA',20),('YUTANDUCHI DE GUERRERO',20),('ZAPOTITLÁN DEL RÍO',20),  
('ZAPOTITLÁN LAGUNAS',20),('ZAPOTITLÁN PALMAS',20),('ZIMATLÁN DE ÁLVAREZ',20),  
('ACAJETE',21),('ACATENO',21),('ACATLÁN',21),('ACATZINGO',21),('ACTEOPAN',21),  
('AHUATLÁN',21),('AHUAZOTEPEC',21),('AHUEHUETITLA',21),('AJALPAN',21),  
('ALBINO ZERTUCHE',21),('ALJOJUCA',21),('ALTEPEXI',21),('AMIXTLÁN',21),('AMOZOC',21),  
('AQUIXTLA',21),('ATEMPAN',21),('ATEXCAL',21),('ATLEQUIZAYAN',21),('ATLIXCO',21),  
('ATOYATEMPAN',21),('ATZALA',21),('ATZITZIHUACÁN',21),('ATZITZINTLA',21),  
('AXUTLA',21),('AYOTOXCO DE GUERRERO',21),('CAÑADA MORELOS',21),('CALPAN',21),  
('CALTEPEC',21),('CAMOCUAUTLA',21),('CAXHUACAN',21),  
('CHALCHICOMULA DE SESMA',21),('CHAPULCO',21),('CHIAUTLA',21),('CHIAUTZINGO',21),  
('CHICHQUILA',21),('CHICONCUAUTLA',21),('CHIETLA',21),('CHIGMECATITLÁN',21),  
('CHIGNAHUAPAN',21),('CHIGNAUTLA',21),('CHILA',21),('CHILA DE LA SAL',21),  
('CHILCHOTLA',21),('CHINANTLA',21),('COATEPEC',21),('COATZINGO',21),  
('COHETZALA',21),('COHUECAN',21),('CORONANGO',21),('COXCATLÁN',21),  
('COYOMEAPAN',21),('COYOTEPEC',21),('CUAPIAXTLA DE MADERO',21),  
('CUAUTEMPAN',21),('CUAUTINCHÁN',21),('CUAUTLANCINGO',21),  
('CUAYUCA DE ANDRADE',21),('CUETZALAN DEL PROGRESO',21),('CUYOACO',21),  
('DOMINGO ARENAS',21),('ELOXOCHITLÁN',21),('EPATLÁN',21),('ESPERANZA',21),  
('FRANCISCO Z. MENA',21),('GENERAL FELIPE ÁNGELES',21),  
('GUADALUPE',21),('GUADALUPE VICTORIA',21),('HERMENEGILDO GALEANA',21),  
('HONEY',21),('HUAQUECHULA',21),('HUATLATLAUCA',21),('HUAUCHINANGO',21),  
('HUEHUETLA',21),('HUEHUETLÁN EL CHICO',21),('HUEHUETLÁN EL GRANDE',21),  
('HUEJOTZINGO',21),('HUEYAPAN',21),('HUEYTAMALCO',21),('HUEYTLALPAN',21),  
('HUITZILAN DE SERDÁN',21),('HUITZILTEPEC',21),('IXCAMILPA DE GUERRERO',21),  
('IXCAQUIXTLA',21),('IXTACAMAXTITLÁN',21),('IXTEPEC',21),  
('IZÚCAR DE ATAMOROS',21),('JALPAN',21),('JOLALPAN',21),('JONOTLA',21),('JOPALA',21),  
('JUAN C. BONILLA',21),('JUAN GALINDO',21),('JUAN N. MÉNDEZ',21),  
('LA MAGDALENA TLATLAUQUITEPEC',21),('LAFRAGUA',21),('LIBRES',21),  
('LOS REYES DE JUÁREZ',21),('MAZAPILTEPEC DE JUÁREZ',21),('MIXTLA',21),  
('MOLCAXAC',21),('NAUPAN',21),('NAUZONTLA',21),('NEALTICAN',21),  
('NICOLÁS BRAVO',21),('NOPALUCAN',21),('OCOTEPEC',21),('OCOYUCAN',21),  
('OLINTLA',21),('ORIENTAL',21),('PAHUATLÁN',21),('PALMAR DE BRAVO',21),  
('PANTEPEC',21),('PETLALCINGO',21),('PIAXTLA',21),('PUEBLA',21),  
('QUECHOLAC',21),('QUIMIXTLÁN',21),('RAFAEL LARA GRAJALES',21),  
('SAN ANDRÉS CHOLULA',21),('SAN ANTONIO CAÑADA',21),  
('SAN DIEGO LA MESA TOCHIMILTZINGO',21),('SAN FELIPE TEOTLALCINGO',21),  
('SAN FELIPE TEPATLÁN',21),('SAN GABRIEL CHILAC',21),  
('SAN GREGORIO ATZOMPA',21),('SAN JERÓNIMO TECUANIPAN',21),  
('SAN JERÓNIMO XAYACATLÁN',21),('SAN JOSÉ CHIAPA',21),  
('SAN JOSÉ MIAHUATLÁN',21),('SAN JUAN ATENCO',21),('SAN JUAN ATZOMPA',21),

('SAN MARTÍN TEXMELUCAN',21),('SAN MARTÍN TOTOLTEPEC',21),  
('SAN MATÍAS TLALANCALECA',21),('SAN MIGUEL IXITLÁN',21),  
('SAN MIGUEL XOXTLA',21),('SAN NICOLÁS BUENOS AIRES',21),  
('SAN NICOLÁS DE LOS RANCHOS',21),('SAN PABLO ANICANO',21),  
('SAN PEDRO CHOLULA',21),('SAN PEDRO YELOIXTLAHUACA',21),  
('SAN SALVADOR EL SECO',21),('SAN SALVADOR EL VERDE',21),  
('SAN SALVADOR HUIXCOLOTLA',21),('SAN SEBASTIÁN TLACOTEPEC',21),  
('SANTA CATARINA TLALTEMPAN',21),('SANTA INÉS AHUATEMPAN',21),  
('SANTA ISABEL CHOLULA',21),('SANTIAGO MIAHUATLÁN',21),  
('SANTO TOMÁS HUEYOTLIPAN',21),('SOLTEPEC',21),('TECALI DE HERRERA',21),  
('TECAMACHALCO',21),('TECOMATLÁN',21),('TEHUACÁN',21),  
('TEHUITZINGO',21),('TENAMPULCO',21),('TEOPANTLÁN',21),('TEOTLALCO',21),  
('TEPANCO DE LÓPEZ',21),('TEPANGO DE RODRÍGUEZ',21),  
('TEPATLAXCO DE HIDALGO',21),('TEPEACA',21),('TEPEMAXALCO',21),('TEPEOJUMA',21),  
('TEPETZINTLA',21),('TEPEXCO',21),('TEPEXI DE RODRÍGUEZ',21),  
('TEPEYAHUALCO',21),('TEPEYAHUALCO DE CUAUHTÉMOC',21),  
('TETELA DE OCAMPO',21),('TETELES DE AVILA CASTILLO',21),('TEZIUTLÁN',21),  
('TIANGUISMANALCO',21),('TILAPA',21),('TLACHICHUCA',21),  
('TLACOTEPEC DE BENITO JUÁREZ',21),('TLACUILOTEPEC',21),('TLAHUAPAN',21),  
('TLALTENANGO',21),('TLANEPANTLA',21),('TLAOLA',21),('TLAPACOYA',21),  
('TLAPANALÁ',21),('TLATLAUQUITEPEC',21),('TLAXCO',21),('TOCHIMILCO',21),  
('TOCHTEPEC',21),('TOTOLTEPEC DE GUERRERO',21),('TULCINGO',21),  
('TUZAMAPAN DE GALEANA',21),('TZICATLACOYAN',21),  
('VENUSTIANO CARRANZA',21),('VICENTE GUERRERO',21),  
('XAYACATLÁN DE BRAVO',21),('XICOTEPEC',21),('XICOTLÁN',21),('XIUTETELCO',21),  
('XOCHIAPULCO',21),('XOCHILTEPEC',21),('XOCHITLÁN DE VICENTE SUÁREZ',21),  
('XOCHITLÁN TODOS SANTOS',21),('YAONÁHUAC',21),('YEHUALTEPEC',21),  
('ZACAPALA',21),('ZACAPOAXTLA',21),('ZACATLÁN',21),('ZAPOTITLÁN',21),  
('ZAPOTITLÁN DE MÉNDEZ',21),('ZARAGOZA',21),('ZAUTLA',21),('ZIHUATEUTLA',21),  
('ZINACATEPEC',21),('ZONGOZOTLA',21),('ZOQUIAPAN',21),  
('ZOQUITLÁN',21),('AMEALCO DE BONFIL',22),('ARROYO SECO',22),  
('CADEREYTA DE MONTES',22),('COLÓN',22),('CORREGIDORA',22),  
('EZEQUIEL MONTES',22),('HUIMILPAN',22),('JALPAN DE SERRA',22),  
('LANDA DE MATAMOROS',22),('PEÑAMILLER',22),('PEDRO ESCOBEDO',22),  
('PINAL DE AMOLES',22),('QUERÉTARO',22),('SAN JOAQUÍN',22),('SAN JUAN DEL RÍO',22),  
('TEQUISQUIAPAN',22),('TOLIMÁN',22),('BENITO JUÁREZ',23),('COZUMEL',23),  
('FELIPE CARRILLO PUERTO',23),('ISLA MUJERES',23),('JOSÉ MARÍA MORELOS',23),  
('OTHÓN P. BLANCO',23),('SOLIDARIDAD',23),  
('AHUALULCO',24),('ALAUQUINES',24),('AQUISMÓN',24),('ARMADILLO DE LOS INFANTE',24),  
('AXTLA DE TERRAZAS',24),('CÁRDENAS',24),('CEDRAL',24),('CERRITOS',24),

('CERRO DE SAN PEDRO',24),('CHARCAS',24),('CIUDAD DEL MAÍZ',24),  
('CIUDAD FERNÁNDEZ',24),('CIUDAD VALLES',24),('COXCATLÁN',24),('EBANO',24),  
('EL NARANJO',24),('GUADALCÁZAR',24),('HUEHUETLÁN',24),('LAGUNILLAS',24),  
('MATEHUALA',24),('MATLAPA',24),('MEXQUITIC DE CARMONA',24),('MOCTEZUMA',24),  
('RAYÓN',24),('RIOVERDE',24),('SALINAS',24),('SAN ANTONIO',24),  
('SAN CIRO DE ACOSTA',24),('SAN LUIS POTOSÍ',24),  
('SAN MARTÍN CHALCHICUAUTLA',24),('SAN NICOLÁS TOLENTINO',24),  
('SAN VICENTE TANCUAYALAB',24),('SANTA CATARINA',24),('SANTA MARÍA DEL RÍO',24),  
('SANTO DOMINGO',24),('SOLEDAD DE GRACIANO SÁNCHEZ',24),('TAMASOPO',24),  
('TAMAZUNCHALE',24),('TAMPACÁN',24),('TAMPAMOLÓN CORONA',24),  
('TAMUÍN',24),('TANCANHUITZ',24),('TANLAJÁS',24),('TANQUIÁN DE ESCOBEDO',24),  
('TIERRA NUEVA',24),('VANEGAS',24),('VENADO',24),('VILLA DE ARISTA',24),  
('VILLA DE ARRIAGA',24),('VILLA DE GUADALUPE',24),('VILLA DE LA PAZ',24),  
('VILLA DE RAMOS',24),('VILLA DE REYES',24),('VILLA HIDALGO',24),('VILLA JUÁREZ',24),  
('XILITLA',24),('ZARAGOZA',24),('AHOME',25),('ANGOSTURA',25),('BADIRAGUATO',25),  
('CHOIX',25),('CONCORDIA',25),('CULIACÁN',25),('EL FUERTE',25),('ELOTA',25),  
('ESCUINAPA',25),('GUASAVE',25),('MAZATLÁN',25),('MOCORITO',25),('NAVOLATO',25),  
('ROSARIO',25),('SALVADOR ALVARADO',25),('SAN IGNACIO',25),('SINALOA',25),  
('ACONCHI',26),('AGUA PRIETA',26),('ALAMOS',26),('ALTAR',26),('ARIVECHI',26),  
('ATIL',26),('BACADÉHUACHI',26),('BACANORA',26),('BACERAC',26),('BACOACHI',26),  
('BANÁMICHÍ',26),('BAVIÁCORA',26),('BAVISPE',26),('BÁCUM',26),('BENITO JUÁREZ',26),  
('BENJAMÍN HILL',26),('CABORCA',26),('CABORCA',26),('CANANEA',26),('CARBÓ',26),  
('CUCURPE',26),('CUMPAS',26),('DIVISADEROS',26),('EMPALME',26),('ETCHOJOA',26),  
('FRONTERAS',26),('GENERAL PLUTARCO ELÍAS CALLES',26),('GRANADOS',26),  
('GUAYMAS',26),('HERMOSILLO',26),('HUACHINERA',26),('HUATABAMPO',26),  
('HUÁSABAS',26),('HUÉPAC',26),('IMURIS',26),('LA COLORADA',26),('MAGDALENA',26),  
('MAZATÁN',26),('MOCTEZUMA',26),('NACO',26),('NACUZARI DE GARCÍA',26),  
('NAVOJOA',26),('NÁCORI CHICO',26),('NOGALES',26),('ONAVAS',26),  
('OPODEPE',26),('OQUITOA',26),('PITIQUITO',26),('PUERTO PEÑASCO',26),('QUIRIEGO',26),  
('RAYÓN',26),('ROSARIO',26),('SAHUARIPA',26),('SAN FELIPE DE JESÚS',26),  
('SAN IGNACIO RÍO MUERTO',26),('SAN JAVIER',26),('SAN LUIS RÍO COLORADO',26),  
('SAN MIGUEL DE HORCASITAS',26),('SAN PEDRO DE LA CUEVA',26),('SANTA ANA',26),  
('SANTA CRUZ',26),('SÁRIC',26),('SOYOPA',26),('SUAQUI GRANDE',26),('TEPACHE',26),  
('TRINCHERAS',26),('TUBUTAMA',26),('URES',26),('VILLA HIDALGO',26),  
('VILLA PESQUEIRA',26),('YÉCORÁ',26),('BALANCÁN',27),('CÁRDENAS',27),  
('CENTLA',27),('CENTRO',27),('COMALCALCO',27),('EMILIANO ZAPATA',27),  
('HUIMANGUILLO',27),('JALAPA',27),('JALPA DE MÉNDEZ',27),('JONUTA',27),  
('MACUSPANA',27),('NACAJUCA',27),('PARAÍSO',27),('TACOTALPA',27),('TEAPA',27),  
('TENOSIQUE',27),('ABASOLO',28),('ALDAMA',28),('ALTAMIRA',28),  
('ANTIGUO MORELOS',28),('BURGOS',28),('CAMARGO',28),('CASAS',28),

('CIUDAD MADERO',28),('CRUILLAS',28),('EL MANTE',28),  
('GÓMEZ FARÍAS',28),('GÜÉMEZ',28),('GONZÁLEZ',28),('GUERRERO',28),  
('GUSTAVO DÍAZ ORDAZ',28),('HIDALGO',28),('JAUMAVE',28),('JIMÉNEZ',28),('LLERA',28),  
('MAINERO',28),('MATAMOROS',28),('MÉNDEZ',28),('MIER',28),('MIGUEL ALEMÁN',28),  
('MIQUIHUANA',28),('NUEVO LAREDO',28),('NUEVO MORELOS',28),('OCAMPO',28),  
('PADILLA',28),('PALMILLAS',28),('RÍO BRAVO',28),('REYNOSA',28),  
('SAN CARLOS',28),('SAN FERNANDO',28),('SAN NICOLÁS',28),  
('SOTO LA MARINA',28),('TAMPICO',28),('TULA',28),('VALLE',28),('VICTORIA',28),  
('VILLAGRÁN',28),('XICOTÉNCATL',28),('ACUAMANALA DE MIGUEL HIDALGO',29),  
('ALTZAYANCA',29),('AMAXAC DE GUERRERO',29),  
('APETATILÁN DE ANTONIO CARVAJAL',29),('APIZACO',29),('BENITO JUÁREZ',29),  
('CALPULALPAN',29),('CHIAUTEMPAN',29),('CONTLA DE JUAN CUAMATZI',29),  
('CUAPIAXTLA',29),('CUAXOMULCO',29),('EL CARMEN TEQUEXQUITLA',29),  
('EMILIANO ZAPATA',29),('ESPAÑITA',29),('HUAMANTLA',29),  
('HUEYOTLIPAN',29),('IXTACUIXTLA DE MARIANO MATAMOROS',29),('IXTENCO',29),  
('LA MAGDALENA TLALTELULCO',29),('LÁZARO CÁRDENAS',29),  
('MAZATECOCHCO DE JOSÉ MARÍA MORELOS',29),('MUÑOZ DE DOMINGO ARENAS',29),  
('NANACAMILPA DE MARIANO ARISTA',29),('NATÍVITAS',29),('PANOTLA',29),  
('PAPALOTLA DE XICOHTÉNCATL',29),('SAN DAMIÁN TEXOLOC',29),  
('SAN FRANCISCO TETLANOHCAN',29),('SAN JERÓNIMO ZACUALPAN',29),  
('SAN JOSÉ TEACALCO',29),('SAN JUAN HUACTZINCO',29),  
('SAN LORENZO AXOCOMANITLA',29),('SAN LUCAS TECOPILCO',29),  
('SAN PABLO DEL MONTE',29),('SANCTÓRUM DE LÁZARO CÁRDENAS',29),  
('SANTA ANA NOPALUCAN',29),('SANTA APOLONIA TEACALCO',29),  
('SANTA CATARINA AYOMETLA',29),('SANTA CRUZ QUILEHTLA',29),  
('SANTA CRUZ TLAXCALA',29),('SANTA ISABEL XILOXOXTLA',29),('TENANCINGO',29),  
('TEOLOCHOLCO',29),('TEPETITLA DE LARDIZÁBAL',29),('TEPEYANCO',29),  
('TERRENATE',29),('TETLA DE LA SOLIDARIDAD',29),('TETLATLAHUCA',29),  
('TLAXCALA',29),('TLAXCO',29),('TOCATLÁN',29),('TOTOLAC',29),('TZOMPANTEPEC',29),  
('XALOZTOC',29),('XALTOCAN',29),('XICOHTZINCO',29),('YAUHQUEMECAN',29),  
('ZACATELCO',29),('ZITLALTEPEC DE TRINIDAD SÁNCHEZ SANTOS',29),  
('ACAJETE',30),('ACATLÁN',30),('ACAYUCAN',30),('ACTOPAN',30),('ACULA',30),  
('AGUA DULCE',30),('ALPATLÁHUAC',30),('ALTO LUCERO DE GUTIÉRREZ BARRIOS',30),  
('ALTOTONGA',30),('ALVARADO',30),('AMATITLÁN',30),('AMATLÁN DE LOS REYES',30),  
('ANGEL R. CABADA',30),('APAZAPAN',30),('AQUILA',30),('ASTACINGA',30),  
('ATLAHUILCO',30),('ATOYAC',30),('ATZACAN',30),('ATZALAN',30),('AYAHUALULCO',30),  
('BANDERILLA',30),('BENITO JUÁREZ',30),('BOCA DEL RÍO',30),('CALCAHUALCO',30),  
('CAMARÓN DE TEJEDA',30),('CAMERINO Z. MENDOZA',30),('CARLOS A. CARRILLO',30),  
('CARRILLO PUERTO',30),('CASTILLO DE TEAYO',30),('CATEMACO',30),('CAZONES',30),

('CÓRDOBA',30),('CERRO AZUL',30),('CHACALTIANGUIS',30),('CHALMA',30),  
('CHICONAMEL',30),('CHICONQUIACO',30),('CHICONTEPEC',30),('CHINAMECA',30),  
('CHINAMPA DE GOROSTIZA',30),('CHOCAMÁN',30),('CHONTLA',30),('CHUMATLÁN',30),  
('CITLALTÉPETL',30),('COACOATZINTLA',30),('COAHUITLÁN',30),('COATEPEC',30),  
('COATZACOALCOS',30),('COATZINTLA',30),('COETZALA',30),('COLIPA',30),('COMAPA',30),  
('COSAMALOAPAN DE CARPIO',30),('COSAUTLÁN DE CARVAJAL',30),  
('COSCOMATEPEC',30),('COSOLEACAQUE',30),('COTAXTLA',30),('COXQUIHUI',30),  
('COYUTLA',30),('CUICHAPA',30),('CUITLÁHUAC',30),('EL HIGO',30),('EMILIANO  
ZAPATA',30),('ESPINAL',30),('FILOMENO MATA',30),('FORTÍN',30),  
('GUTIÉRREZ ZAMORA',30),('HIDALGOTITLÁN',30),('HUATUSCO',30),  
('HUAYACOCOTLA',30),('HUEYAPAN DE OCAMPO',30),('HUILOAPAN',30),  
('IGNACIO DE LA LLAVE',30),('ILAMATLÁN',30),('ISLA',30),('IXCATEPEC',30),  
('IXHUACÁN DE LOS REYES',30),('IXHUATLANCILLO',30),('IXHUATLÁN DE MADERO',30),  
('IXHUATLÁN DEL CAFÉ',30),('IXHUATLÁN DEL SURESTE',30),('IXMATLAHUACAN',30),  
('IXTACZOQUITLÁN',30),('JALACINGO',30),('JALCOMULCO',30),('JAMAPA',30),  
('JÁLTIPAN',30),('JESÚS CARRANZA',30),('JILOTEPEC',30),('JOSÉ AZUETA',30),  
('JUAN RODRÍGUEZ CLARA',30),('JUCHIQUE DE FERRER',30),('LA ANTIGUA',30),  
('LA PERLA',30),('LANDERO Y COSS',30),('LAS CHOAPAS',30),('LAS MINAS',30),  
('LAS VIGAS DE RAMÍREZ',30),('LERDO DE TEJADA',30),  
('LOS REYES',30),('MAGDALENA',30),('MALTRATA',30),('MANLIO FABIO ALTAMIRANO',30),  
('MARIANO ESCOBEDO',30),('MARTÍNEZ DE LA TORRE',30),('MECATLÁN',30),  
('MECAYAPAN',30),('MEDELLÍN',30),('MIAHUATLÁN',30),('MINATITLÁN',30),('MISANTLA',30),  
('MIXTLA DE ALTAMIRANO',30),('MOLOACÁN',30),  
('NANCHITAL DE LÁZARO CÁRDENAS DEL RÍO',30),('NAOLINCO',30),('NARANJAL',30),  
('NARANJOS AMATLÁN',30),('NAUTLA',30),('NOGALES',30),('OLUTA',30),('OMEALCA',30),  
('ORIZABA',30),('OTATITLÁN',30),('OTEAPAN',30),  
('OZULUAMA DE MASCAREÑAS',30),('PAJAPAN',30),('PAPANTLA',30),  
('PASO DE OVEJAS',30),('PASO DEL MACHO',30),('PÁNUCO',30),('PEROTE',30),  
('PLATÓN SÁNCHEZ',30),('PLAYA VICENTE',30),('POZA RICA DE HIDALGO',30),  
('PUEBLO VIEJO',30),('PUENTE NACIONAL',30),('RAFAEL DELGADO',30),  
('RAFAEL LUCIO',30),('RÍO BLANCO',30),('SALTABARRANCA',30),  
('SAN ANDRÉS TENEJAPAN',30),('SAN ANDRÉS TUXTLA',30),  
('SAN JUAN EVANGELISTA',30),('SAN RAFAEL',30),('SANTIAGO SOCHIAPAN',30),  
('SANTIAGO TUXTLA',30),('SAYULA DE ALEMÁN',30),('SOCHIAPA',30),('SOCONUSCO',30),  
('SOLEDAD ATZOMPA',30),('SOLEDAD DE DOBLADO',30),('SOTEAPAN',30),  
('TAMALÁN',30),('TAMIAHUA',30),('TAMPICO ALTO',30),('TANCOCO',30),('TANTIMA',30),  
('TANTOYUCA',30),('TATAHUICAPAN DE JUÁREZ',30),('TATATILA',30),('TÚXPAM',30),  
('TECOLUTLA',30),('TEHUIPANGO',30),('TEMAPACHE',30),('TEMPOAL',30),('TENAMPA',30),  
('TENOCHTITLÁN',30),('TEOCELO',30),('TEPATLAXCO',30),('TEPETLÁN',30),  
('TEPETZINTLA',30),('TEQUILA',30),('TEXCATEPEC',30),('TEXHUACÁN',30),

('TEXISTEPEC',30),('TEZONAPA',30),('TIERRA BLANCA',30),('TIHUATLÁN',30),  
 ('TLACHICHILCO',30),('TLACOJALPAN',30),('TLACOLULAN',30),('TLACOTALPAN',30),  
 ('TLACOTEPEC DE MEJÍA',30),('TLALIXCOYAN',30),('TLALNELHUAYOCAN',30),  
 ('TLALTETELA',30),('TLAPACOYAN',30),('TLAQUILPA',30),('TLILAPAN',30),  
 ('TOMATLÁN',30),('TONAYÁN',30),('TOTUTLA',30),('TRES VALLES',30),('TUXTILLA',30),  
 ('URSULO GALVÁN',30),('UXPANAPA',30),('VEGA DE ALATORRE',30),('VERACRUZ',30),  
 ('VILLA ALDAMA',30),('XALAPA',30),('XICO',30),('XOXOCOTLA',30),('YANGA',30),  
 ('YECUATLA',30),('ZACUALPAN',30),('ZARAGOZA',30),('ZENTLA',30),('ZONGOLICA',30),  
 ('ZONTECOMATLÁN DE LÓPEZ Y FUENTES',30),('ZOZOCOLCO DE HIDALGO',30),  
 ('ABALÁ',31),('ACANCEH',31),('AKIL',31),('BACA',31),('BOKOBÁ',31),('CACALCHÉN',31),  
 ('CALOTMUL',31),('CANSAHCAB',31),('CANTAMAYEC',31),('CELESTÚN',31),  
 ('CENOTILLO',31),('CHACSINKÍN',31),('CHANKOM',31),('CHAPAB',31),('CHEMAX',31),  
 ('CHICHIMILÁ',31),('CHICXULUB PUEBLO',31),('CHIKINDZONOT',31),('CHOCHOLÁ',31),  
 ('CHUMAYEL',31),('CONKAL',31),('CUNCUNUL',31),('CUZAMÁ',31),('DZÁN',31),  
 ('DZEMUL',31),('DZIDZANTÚN',31),('DZILAM DE BRAVO',31),('DZILAM GONZÁLEZ',31),  
 ('DZITÁS',31),('DZONCAUICH',31),('ESPITA',31),('HALACHÓ',31),('HOCABÁ',31),  
 ('HOCTÚN',31),('HOMÚN',31),('HUHÍ',31),('HUNUCMÁ',31),('IXIL',31),('IZAMAL',31),  
 ('KANASÍN',31),('KANTUNIL',31),('KAUA',31),('KINCHIL',31),('KOPOMÁ',31),('MAMA',31),  
 ('MANÍ',31),('MAXCANÚ',31),('MAYAPÁN',31),('MÉRIDA',31),('MOCOCHÁ',31),  
 ('MOTUL',31),('MUNA',31),('MUXUPIP',31),('OPICHÉN',31),('OXKUTZCAB',31),  
 ('PANABÁ',31),('PETO',31),('PROGRESO',31),('QUINTANA ROO',31),('RÍO LAGARTOS',31),  
 ('SACALUM',31),('SAMAHIL',31),('SAN FELIPE',31),('SANAHCAT',31),('SANTA ELENA',31),  
 ('SEYÉ',31),('SINANCHÉ',31),('SOTUTA',31),('SUCILÁ',31),('SUDZAL',31),  
 ('SUMA',31),('TAHDZIÚ',31),('TAHMEK',31),('TEABO',31),('TECOH',31),  
 ('TEKAL DE VENEGAS',31),('TEKANTÓ',31),('TEKAX',31),('TEKIT',31),  
 ('TEKOM',31),('TELCHAC PUEBLO',31),('TELCHAC PUERTO',31),('TEMAX',31),  
 ('TEMOZÓN',31),('TEPAKÁN',31),('TETIZ',31),('TEYA',31),('TICUL',31),('TIMUCUY',31),  
 ('TINUM',31),('TIXCACALCUPUL',31),('TIXKOKOB',31),('TIXMEHUAC',31),('TIXPÉHUAL',31),  
 ('TIZIMÍN',31),('TUNKÁS',31),('TZUCACAB',31),('UAYMA',31),('UCÚ',31),('UMÁN',31),  
 ('VALLADOLID',31),('XOCHEL',31),('YAXCABÁ',31),('YAXKUKUL',31),('YOBAÍN',31),  
 ('APOZOL',32),('APULCO',32),('ATOLINGA',32),('BENITO JUÁREZ',32),  
 ('CAÑITAS DE FELIPE PESCADOR',32),('CHALCHIHUITES',32),  
 ('CONCEPCIÓN DEL ORO',32),('CUAUHTÉMOC',32),  
 ('EL PLATEADO DE JOAQUÍN AMARO',32),('EL SALVADOR',32),('FRESNILLO',32),  
 ('GENARO CODINA',32),('GENERAL ENRIQUE ESTRADA',32),  
 ('GENERAL FRANCISCO R. MURGUÍA',32),('GENERAL PÁNFILO NATERA',32),  
 ('GUADALUPE',32),('HUANUSCO',32),('JALPA',32),('JEREZ',32),('JIMÉNEZ DEL TEUL',32),  
 ('JUAN ALDAMA',32),('JUCHIPILA',32),('LORETO',32),('LUIS MOYA',32),('MAZAPIL',32),  
 ('MELCHOR OCAMPO',32),('MEZQUITAL DEL ORO',32),('MIGUEL AUZA',32),('MOMAX',32),  
 ('MONTE ESCOBEDO',32),('MORELOS',32),('MOYAHUA DE ESTRADA',32),

('NOCHISTLÁN DE MEJÍA',32),('NORIA DE ÁNGELES',32),('OJOCALIENTE',32),  
('PÁNUCO',32),('PINOS',32),('RÍO GRANDE',32),('SAIN ALTO',32),  
('SANTA MARÍA DE LA PAZ',32),('SOMBRETE',32),('SUSTICACÁN',32),('TABASCO',32),  
('TEPECHTLÁN',32),('TEPETONGO',32),('TEUL DE GONZÁLEZ ORTEGA',32),  
('TLALTENANGO DE SÁNCHEZ ROMÁN',32),('TRANCOSO',32),  
('TRINIDAD GARCÍA DE LA CADENA',32),('VALPARAÍSO',32),('VETAGRANDE',32),  
('VILLA DE COS',32), ('VILLA GARCÍA',32),('VILLA GONZÁLEZ ORTEGA',32),  
('VILLA HIDALGO',32),('VILLANUEVA',32),('ZACATECAS',32);

INSERT INTO CAT\_NIVELES\_SOCIOECONOMICOS(NIVEL\_SOCIOECONOMICO)  
VALUES('A/B CLASE RICA'),('C+ CLASE MEDIA ALTA'),('C CLASE MEDIA'),  
('D+ CLASE MEDIA BAJA'),('D CLASE POBRE'),('E POBREZA EXTREMA');

INSERT INTO CAT\_VIVIENDAS(VIVIENDA)VALUES  
('CASA BIOCLIMÁTICA O BIOENERGÉTICA'), ('CASA DE LADRILLO'),  
('CASA DE MATERIALES MIXTOS'),('CASA DE PAJA, RAMAS O CAÑAS');

INSERT INTO CAT\_CLUES(CLUES,DESCRIPCION)VALUES ('S/C','SIN CLUES'),  
('CRO','CRUZ ROJA MEXICANA'),('HUN','HOSPITALES UNIVERSITARIOS'),  
('IST','INSTITUTO DE SEGURIDAD Y SERVICIOS SOCIALES PARA LOS TRABAJADORES  
DEL ESTADO'),('IMS','INSTITUTO MEXICANO DEL SEGURO SOCIAL'),  
('IMO','INSTITUTO MEXICANO DEL SEGURO SOCIAL. REGIMEN OPORTUNIDADES'),  
('PMX','PETROLEOS MEXICANOS'),('SDN','SECRETARÍA DE LA DEFENSA NACIONAL'),  
('SMA','SECRETARÍA DE MARINA'),('SSA','SECRETARÍA DE SALUD'),  
('SME','SERVICIOS MÉDICOS ESTATALES'),('SMM','SERVICIOS MÉDICOS MUNICIPALES'),  
('SMP','SERVICIOS MÉDICOS PRIVADOS'),  
('DIF','SISTEMA NACIONAL PARA EL DESARROLLO INTEGRAL DE LA FAMILIA');

INSERT INTO CAT\_MODULOS(MODULO)VALUES('SISTEMA'),('CONTROL DE USUARIOS'),  
('CONTROL DE PACIENTES'),('DATOS DE SANIDAD'),('CREDENCIALIZACION'),  
('RECOLECCION');

INSERT INTO CAT\_MOVIMIENTOS(MOVIMIENTO)VALUES('ENTRO'),('SALIO'),('AGREGO'),  
('ELIMINO'),('MODIFICO');

```

INSERT INTO CAT_ESPECIALIDADES(ESPECIALIDAD)VALUES('NINGUNA'),
('ALERGOLOGÍA'),('ALGOLOGÍA'),('ANESTESIOLOGÍA'),
('ANGIOLOGÍA Y CIRUGÍA VASCULAR'),('AUDIOLOGÍA, FONIATRÍA'),('BARIATRÍA'),
('CARDIOLOGÍA'),('CIRUGÍA PLÁSTICA Y RECONSTRUCTIVA'),('COLOPROCTOLOGÍA'),
('DERMATOLOGÍA'),('DERMATOLOGÍA PEDIÁTRICA'),('ENDOCRINOLOGÍA'),
('ENDOSCOPIA'),('ENDODONCIA'),('GASTROENTEROLOGÍA'),('GENÉTICA'),('GERIATRÍA'),
('GERONTOLOGÍA'),('HEMATOLOGÍA'),('HEMATO-ONCOLOGÍA'),('HEPATOLOGÍA'),
('IMAGENOLOGÍA'),('INFECTOLOGÍA'),('INHALOTERAPIA'),
('INMUNOLOGÍA CLÍNICA Y ALERGOLOGÍA PEDIÁTRICA'),
('MEDICINA FÍSICA Y DE REHABILITACIÓN'),('MEDICINA CRÍTICA'),('MEDICINAGENERAL'),
('MEDICINA FAMILIAR'),('MEDICINA DEL DEPORTE'),('MEDICINA NUCLEAR'),
('MEDICINA PREVENTIVA'),('NEFROLOGÍA'),('NEONATOLOGÍA'),('NEUMOLOGÍA'),
('NEUROCIRUGÍA'),('NEURORADIOLOGÍA'),('NEUROFISIOLOGÍA'),('NEUROLOGÍA'),
('NUTRIOLOGÍA'),('ODONTOLOGÍA'),('OFTALMOLOGÍA'),('ONCOLOGÍA'),
('TÉCNICO ORTESISTA'),('ORTOPEDIA'),('OTORRINOLARINGOLOGÍA'),('PATOLOGÍA'),
('PEDIATRÍA'),('PERINATOLOGÍA'),('PROCTOLOGÍA'),('PSICOLOGÍA'),('PSIQUIATRÍA'),
('QUIROPRAXIA'),('RADIOLOGÍA'),('RADIOTERAPIA'),('REHABILITACIÓN PULMONAR'),
('REUMATOLOGÍA'),('TRAUMATOLOGÍA Y ORTOPEDIA'),
('TRAUMATOLOGÍA DEPORTIVA'),('UROLOGÍA');

```

```

INSERT INTO CAT_GRUPOS_ENFERMEDADES(GRUPO_ENFERMEDAD)
VALUES('SISTEMA DIGESTIVO'),('SISTEMA RESPIRATORIO'),('SISTEMA CIRCULATORIO'),
('SISTEMA REPRODUCTOR'),('SISTEMA URINARIO'),('SISTEMA NERVIOSO'),
('HUESOS MÚSCULOS Y ARTICULACIONES'),('OJO Y SUS ANEXOS'),
('OÍDO, GARGANTA Y NARIZ'),('DE LA SANGRE Y INMUNIDAD'),('DE LA PIEL'),
('DE LA NUTRICIÓN, ENDÓCRINAS Y DEL METABOLISMO'),
('INFECCIONES Y PARASITARIAS'),('TRANSMISIÓN SEXUAL'),('REUMATOLOGÍA'),
('TRASTORNOS MENTALES Y DEL COMPORTAMIENTO');

```

```

INSERT INTO CAT_ENFERMEDADES(IDGRUPO_ENFERMEDAD,ENFERMEDAD)VALUES
(1,'CÁNCER DE COLON'),(1,'COLITIS ULCERATIVA'),(1,'DIVERTICULOSIS'),
(1,'ENFERMEDAD DE CROHN'),(1,'PANCREATITIS'),
(1,'SÍNDROME DE INTESTINO IRRITABLE'),(1,'REFLUJO GASTROESOFÁGICO'),(2,'ASMA'),
(2,'CÁNCER DE PULMÓN'),(2,'GRIPE-INFLUENZA'),(3,'ANGINA DE PECHO'),
(3,'ARRITMIA CARDÍACA'),(3,'ATAQUE CARDÍACO'),(3,'FIBRILACIÓN AURICULAR'),
(3,'INSUFICIENCIA CARDÍACA'),(3,'PRESIÓN ARTERIAL ALTA (HIPERTENSIÓN)'),
(3,'PROLAPSO DE LA VÁLVULA MITRAL'),(3,'QUISTES GANGLIONARES'),(3,'VASCULITIS'),
(3,'VENAS VARICOSAS'),(4,'CÁNCER DE LA PRÓSTATA'),(4,'CÁNCER DE OVARIOS'),
(4,'ENDOMETRIOSIS'),(4,'CÁNCER DEL SENO'),(4,'DISFUNCIÓN ERÉCTIL');

```



(4,'CÁNCER DE OVARIOS'),(4,'FIBROMA UTERINO'),(4,'MENOPAUSIA'),  
 (4,'QUISTES OVÁRICOS'),(5,'CÁLCULOS RENALES'),(5,'FALLO RENAL'),  
 (6,'CONVULSIONES Y EPILEPSIA'),(6,'ENFERMEDAD DE PARKINSON'),  
 (6,'LESIONES DE LA MÉDULA ESPINAL'),(6,'MIGRAÑAS'),  
 (6,'NEURALGIA DEL TRIGÉMINIO'), (6,'PARÁLISIS CEREBRAL'),  
 (6,'PARÁLISIS FACIAL DE BELLS'),(6,'TUMORES CEREBRALES'),  
 (7,'CODO DE TENISTA'),(7,'DOLOR DE ESPALDA'),(7,'FIBROSIS QUÍSTICA'),  
 (7,'FRACTURAS Y TORCEDURAS'),(7,'HERNIA INGUINAL'),(7,'HERNIAS DE INCISIÓN'),  
 (7,'HERNIA UMBILICAL'),(7,'LESIONES DEL MANGUITO ROTATORIO'),  
 (7,'OSTEOPOROSIS'),(7,'TRANSTORNOS TEMPOROMANDIBULARES'),(8,'CATARATAS'),  
 (8,'DEGENERACIÓN MACULAR'),(8,'DIABETES-COMPLICACIONES OCULARES'),  
 (8,'GLAUCOMA'),(9,'ACÚFENOS O ZUMBIDO DEL OÍDO'),(9,'OTITIS MEDIA'),  
 (9,'PÉRDIDA DE LA AUDICIÓN'),(10,'LEUCEMIA'),(10,'LUPUS'),(10,'MIASTENIA GRAVE'),  
 (10,'MIELOMA MÚLTIPLE'),(10,'SARCOIDOSIS'),(11,'ACNÉ'),(11,'ALOPECIA'),  
 (11,'CÁNCER DE LA PIEL'),(11,'ENFERMEDAD DE LYME'),(11,'LEISHMANIASIS'),  
 (11,'MELANOMA'),(11,'PSORIASIS'),(11,'QUEMADURAS'),(11,'SARNA'),(11,'VERRUGAS'),  
 (12,'DIABETES-INTRODUCCIÓN'),(12,'DIABETES - CUIDADO DEL PIE'),  
 (12,'DIABETES-PLANIFICACIÓN DE LAS COMIDAS'),(12,'HIPOGLUCEMIA'),(13,'ANTRAX'),  
 (13,'HEPATITIS C'),(13,'HERPES ZOSTER'),(13,'MALARIA'),(13,'MENINGITIS'),  
 (13,'TUBERCULOSIS'),(13,'VIRUELA'),(13,'VIRUS DE EPSTEIN Y MONONUCLEOSIS'),  
 (14,'VIH'),(14,'SÍFILIS'),(14,'GONORREA'),(14,'CLAMIDIA'),(14,'HERPES GENITAL'),  
 (14,'PAPILOMA HUMANO O VPH'),(15,'ASTRITIS REUMATOIDE'),(15,'ARTRITIS'),  
 (15,'ESCLEROSIS LATERAL AMIOTRÓFICA'),(15,'ESCLEROSIS MÚLTIPLE'),(15,'GOTA'),  
 (15,'OSTEOARTRITIS'),(16,'DEPRESIÓN'),(16,'FIBROMIALGIA'),  
 (16,'TRANSTORNOS DEL SUEÑO');

```
INSERT INTO CAT_GRUPOS_DISCAPACIDADES(GRUPO_DISCAPACIDAD)
VALUES('VISUAL'),('FÍSICA'),('MENTAL'),('AUDITIVA');
```

```
INSERT INTO CAT_DISCAPACIDADES(IDGRUPO_DISCAPACIDAD,DISCAPACIDAD)
VALUES(1,'AMBLIOPÍA'),(1,'CEGUERA'),(1,'GLAUCOMA'),(1,'HEMIANOPSIA'),
(1,'RETINOPATÍA'),(1,'RETINOSIS PIGMENTARIA'),(2,'AMIOTROFIA ESPINAL'),
(2,'HUNTINGTON DE COREA'),(2,'ATAXIAS'),(2,'ATAXIA DE FRIEDREICH'),(2,'DISTONÍA'),
(2,'DISTROFIA MUSCULAR'),(2,'ENFERMEDAD DE DUCHENNE'),
(2,'ESCLEROSIS LATERAL AMIO-TRÓFICA'),(2,'ENFERMEDAD DE PARKINSON'),
(2,'ESCLEROSIS EN PLACAS O MÚLTIPLE'),(2,'ESPINA BÍFIDA'),(2,'PARÁLISIS
CEREBRAL'),(2,'SÍNDROME DE GUILLAIN-BARRÉ'),(3,'AUTISMO'),(3,'ALZHEIMER'),
(3,'SÍNDROME DE PRADER-WILLI'),(3,'SÍNDROME DE DOWN'),
(3,'SÍNDROME DEL CROMOSOMA X FRÁGIL'),(4,'HIPOACUSIA'),(4,'SORDERA');
```

```
INSERT INTO CAT_ALERGIAS(ALERGIA)VALUES('AL HUEVO Y LA LECHE'),
('MARISCOS Y PESCADOS'),('CACAHUATES'),('CHOCOLATE'),('ANTIBIÓTICOS'),
('NUECES'), ('TRIGO'),('SULFAMIDAS'),('ANTICONVULSIVOS'),('PENICILINA'),('POLVO'),
('ACAROS'),('INSECTOS'),('HONGOS'),('CASPA ANIMAL'),('POLEN DE PLANTAS'),
('LATEX'),('HUMO'),('ANIMALES DOMÉSTICOS'),('SUSTANCIAS QUÍMICAS'),('NIQUEL');
```

```
INSERT INTO CAT_ROLES(ROL)VALUES('MASTER'),('ADMINISTRADOR'),('MEDICO'),
('OPERADOR'),('PERSONALIZADO');
```

```
INSERT INTO USUARIOS(CEDULA,CURP,NOMBRE,APELLIDO_PATERO,
APELLIDO_MATERNO,IDESPECIALIDAD,IDSUB_ESPECIALIDAD,IDCLUES,
IDROL,USUARIO,CONTRASENA,FECHA_ALTA,FECHA_BAJA,ACTIVADO)VALUES
('000000','AAAA000000AAAAA00','MASTER','AAAAAA','AAAAAA',1,1,1,1,'master0000','mast
er0000','2001-01-01','9999-12-31',TRUE);
```

```
INSERT INTO USUARIOS_MODULOS(IDUSUARIO,IDMODULO)
VALUES(1,2),(1,3),(1,4),(1,5),(1,6);
```

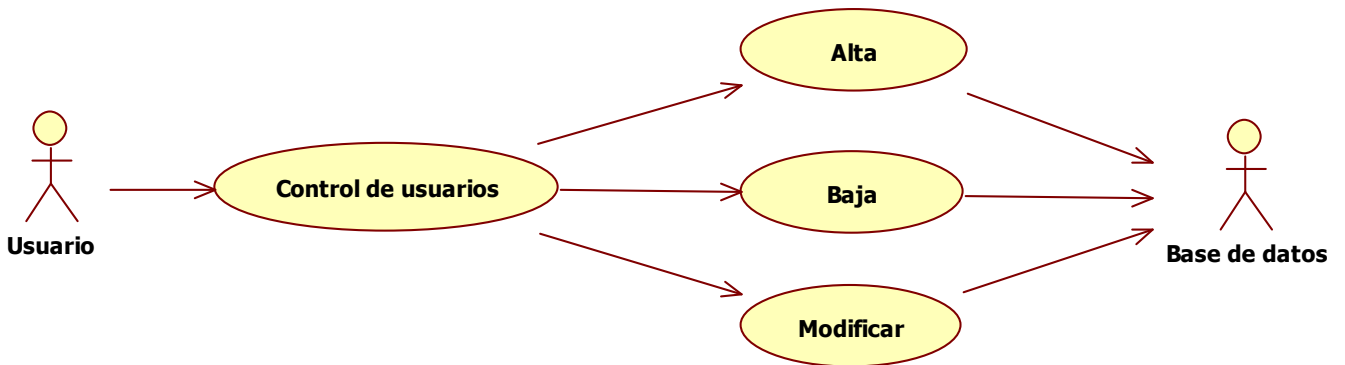
# DISEÑO E IMPLEMENTACIÓN DE HEALTH CARD MANAGER 2010

- Diagramas UML de los módulos de Health Card Manager 2010.

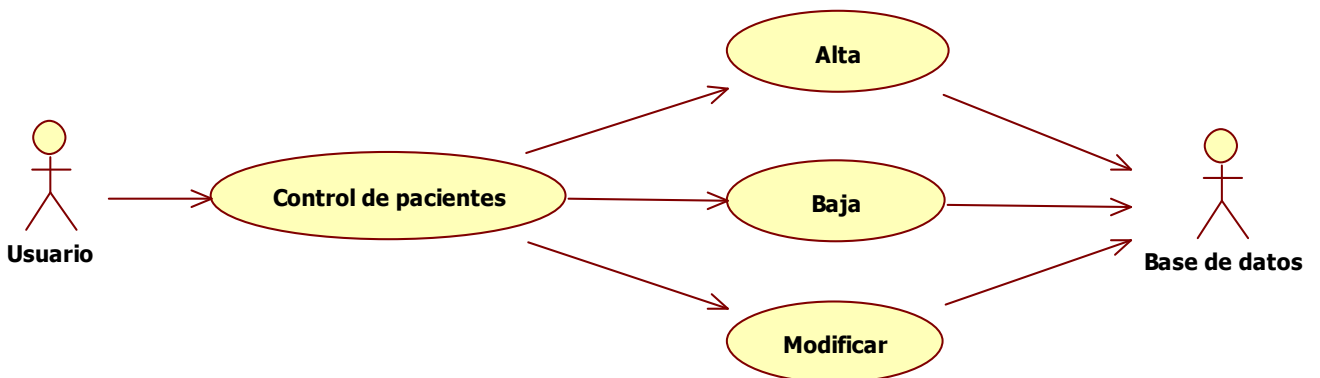
Módulo "Iniciar sesión"



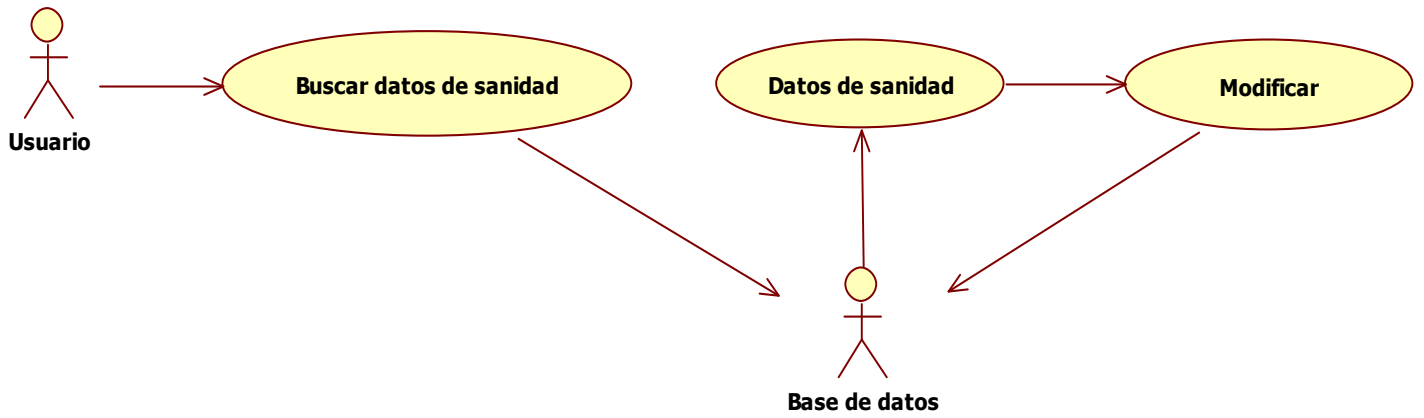
Módulo "Control de usuarios"



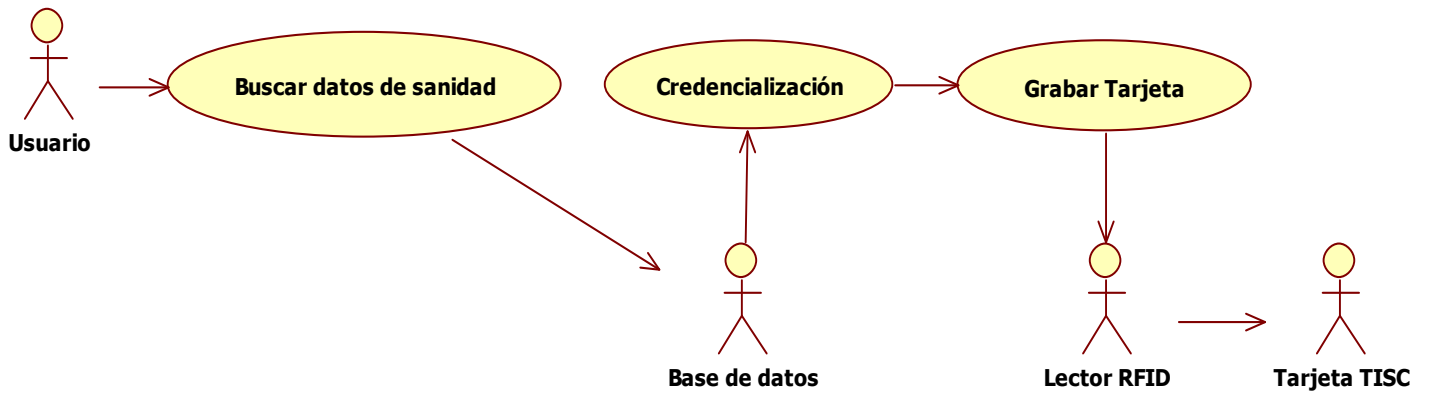
Módulo "Control de pacientes"



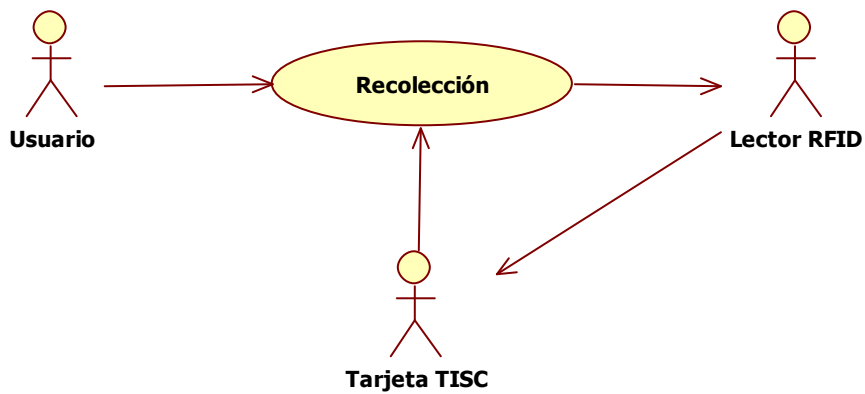
### Módulo "Datos de sanidad"



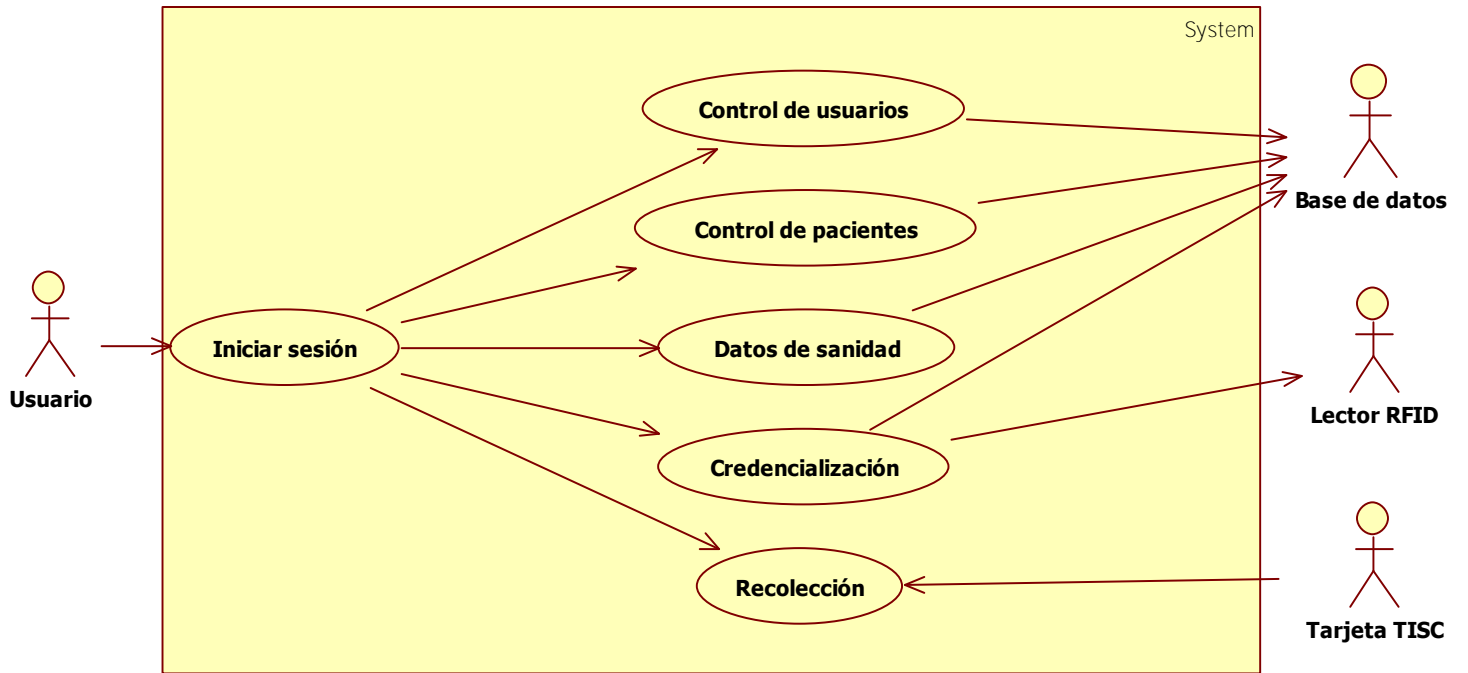
### Módulo "Credencialización"



### Módulo "Recolección"



# Módulo "Interfaz de integración"



- Documentación de actores de Health Card Manager 2010.

<b>Actor</b>	Usuario
<b>Casos</b>	Iniciar sesión, Interfaz de integración, Alta usuario, Baja usuario, Modificar usuario, Alta paciente, Baja paciente, Modificar paciente, Modificar datos de sanidad, Asignar credencial, Recolección.
<b>Tipo</b>	Primario
<b>Descripción</b>	Es el actor principal y representa a cualquier persona que desee utilizar el Health Card Manager 2010.

<b>Actor</b>	Base de datos
<b>Casos</b>	Iniciar sesión, Interfaz de integración, Alta usuario, Baja usuario, Modificar usuario, Alta paciente, Baja paciente, Modificar paciente, Modificar datos de sanidad, Asignar credencial.
<b>Tipo</b>	Secundario
<b>Descripción</b>	Representa una base de datos donde se guarda toda la información relacionada con Health Card Manager 2010.

<b>Actor</b>	Lector RFID
<b>Casos</b>	Asignar credencial, Recolección.
<b>Tipo</b>	Secundario
<b>Descripción</b>	Representa un lector de identificación por radiofrecuencia, lee y graba información en una tarjeta.

<b>Actor</b>	Tarjeta TISC
<b>Casos</b>	Asignar credencial, Recolección.
<b>Tipo</b>	Secundario
<b>Descripción</b>	Representa una tarjeta inteligente sin contacto (mifare classic 4k), almacena 4 Kbytes de información.

- Casos de uso de Health Card Manager 2010.

### Caso de uso “Iniciar sesión”



Pantalla principal (P-1)

<b>Caso de uso</b>	Iniciar sesión
<b>Actores</b>	Usuario, Base de datos
<b>Tipo</b>	Inclusión
<b>Propósito</b>	Validar que el usuario este registrado en Health Card Manager 2010.
<b>Resumen</b>	El usuario ingresara sus datos (usuario y contraseña) para que Health Card Manager 2010 los valide.
<b>Pre condiciones</b>	El usuario debe estar registrado en la base de datos.
<b>Post condiciones</b>	El usuario es validado y se le mostraran la pantalla para que elija otras operaciones.
<b>Flujo principal</b>	Se mostrará la pantalla principal (P-1), el usuario podrá elegir “Entrar” (S-1) o ingresar sus datos (E-1).
<b>Subflujo</b>	(S-1) Health Card Manager 2010 valida los datos ingresados por el usuario (E-2) y se muestra la pantalla (P-2).
<b>Excepciones</b>	(E-1) Si los datos no fueron introducidos será mostrada la misma pantalla (P-1).  (E-2) En caso de uno de los datos no coincida con el almacenado en la base de datos se mostrara el error pertinente.

## Caso de uso "Interfaz de integración"



Pantalla de Interfaz de integración (P-2)

<b>Caso de uso</b>	Interfaz de integración
<b>Actores</b>	Usuario, Base de datos
<b>Tipo</b>	Básico
<b>Propósito</b>	Mostrar al usuario los módulos a los que puede ingresar.
<b>Resumen</b>	El usuario podrá elegir a cuál de los módulos desea ingresar.
<b>Pre condiciones</b>	El usuario tuvo que ser validado en el caso de uso iniciar sesión.
<b>Post condiciones</b>	Se mostrara una pantalla diferente dependiendo la acción.
<b>Flujo principal</b>	Se ejecutara el caso de uso iniciar sesión, posteriormente se mostrara la pantalla (P-2) para el elegir el modulo a ingresar.
<b>Subflujo</b>	(S-1) Control de usuarios. Ingresara a control de usuarios. (S-2) Control de pacientes. Ingresara a control de pacientes. (S-3) Datos de sanidad. Ingresara a datos de sanidad. (S-4) Credencialización. Ingresara a credencialización. (S-4) Recolección. Ingresara a recolección.
<b>Excepciones</b>	Sin Excepciones.



## Caso de uso "Alta usuario"

**Control de usuarios**

Alta

Cédula:  CURP:

Nombre (s):

Apellido paterno:

Apellido materno:

Especialidad:

Subespecialidad:

Rol:  CLUES:

Módulos:

Control de usuarios  Control de pacientes

Datos de sanidad  Credencializacion

Recoleccion

Usuario:

Contraseña:

Pantalla de alta usuario (P-3)

<b>Caso de uso</b>	Alta usuario
<b>Actores</b>	Usuario, Base de datos
<b>Tipo</b>	Básico
<b>Propósito</b>	Ingresar los datos de un nuevo usuario.
<b>Resumen</b>	Registrar un usuario con la información capturada.
<b>Pre condiciones</b>	No debe existir el usuario en la base de datos de HCM 2010.
<b>Post condiciones</b>	Registrar los datos del nuevo usuario en la base de datos.
<b>Flujo principal</b>	Se eligió la opción alta en la pantalla (P-3), se llenan los campos correspondientes y al elegir aceptar se registra en la base de datos.
<b>Subflujo</b>	(S-1) Cancelar. Se regresa a la pantalla (P-3).
<b>Excepciones</b>	(E-1) En caso de no llenar un campo solicitado, se deberá mencionar al usuario que lo ingrese.  (E-2) Verificar que el Usuario nuevo no se encuentre ya registrado.

## Caso de uso "Baja usuario"

Pantalla de baja usuario (P-4)

<b>Caso de uso</b>	Baja usuario
<b>Actores</b>	Usuario, Base de datos
<b>Tipo</b>	Básico
<b>Propósito</b>	Eliminar un usuario registrado.
<b>Resumen</b>	Dar de baja un usuario de la base de datos
<b>Pre condiciones</b>	El usuario debe de estar registrado en la base de datos.
<b>Post condiciones</b>	El usuario aparecerá dado de baja de HCM 2010.
<b>Flujo principal</b>	Se eligió la opción baja en la pantalla (P-3), se elige el usuario y presionamos Eliminar para darlo de baja del HCM 2010.
<b>Subflujo</b>	(S-1) Cancelar. Se regresa a la pantalla (P-3).
<b>Excepciones</b>	(E-1) En caso de que no exista el usuario mostrar el error correspondiente.

## Caso de uso "Modificar usuario"

**Control de usuarios**

**Modificar**

Cédula:  CURP: GORN780217KGMCNF09

Nombre (s): NATIVIDAD

Apellido paterno: GODINEZ

Apellido materno: RAMIREZ

Especialidad:

Subespecialidad:

Rol:  CLUES:

Módulos:

- Control de usuarios
- Control de pacientes
- Datos de sanidad
- Credencializacion
- Recoleccion

Usuario:

Contraseña:

Fecha de alta:  Fecha de baja:

Usuario activado.

Alta Baja Modificar

Aceptar Cancelar

Pantalla de modificar usuario (P-5)

<b>Caso de uso</b>	Modificar usuario
<b>Actores</b>	Usuario, Base de datos
<b>Tipo</b>	Básico
<b>Propósito</b>	Modificar un usuario registrado.
<b>Resumen</b>	Modificar los datos de un usuario.
<b>Pre condiciones</b>	El usuario debe de estar registrado en la base de datos.
<b>Post condiciones</b>	Se almacenaran los cambios realizados en la base de datos.
<b>Flujo principal</b>	Se eligió la opción modificar en la pantalla (P-3), se elige el usuario, se realizan los cambios y presionamos Aceptar para guardar los cambios.
<b>Subflujo</b>	(S-1) Cancelar. Se regresa a la pantalla (P-3).
<b>Excepciones</b>	(E-1) En caso de que no exista el usuario mostrar el error correspondiente. (E-2) Verificar que los datos modificados sean correctos.

## Caso de uso "Alta paciente"

**Control de pacientes**

**Alta**

Numero de paciente: 1101000048

Nombre (s):

Apellido paterno:

Apellido materno:

CURP:

Fecha de nacimiento: 1 Enero 1910

Telefono particular:

Tipo de domicilio: PRIMARIO

Calle:  Número exterior:

Número interior:  Colonia:

Estado: AGUASCALIENTES Telefono de casa:

Municipio: AGUASCALIENTES Código Postal:

Fecha de alta: 2011-1-10 Fecha de baja: 9999-12-31

Alta Baja Modificar

Aceptar Cancelar

Pantalla de alta paciente (P-6)

<b>Caso de uso</b>	Alta paciente
<b>Actores</b>	Usuario, Base de datos
<b>Tipo</b>	Básico
<b>Propósito</b>	Ingresar los datos de un nuevo paciente.
<b>Resumen</b>	Registrar un paciente con la información capturada.
<b>Pre condiciones</b>	No debe existir el paciente en la base de datos de HCM 2010.
<b>Post condiciones</b>	Registrar los datos del nuevo paciente en la base de datos.
<b>Flujo principal</b>	Se eligió la opción alta en la pantalla (P-6), se llenan los campos correspondientes y al elegir aceptar se registra en la base de datos.
<b>Subflujo</b>	(S-1) Cancelar. Se regresa a la pantalla (P-6).
<b>Excepciones</b>	(E-1) En caso de no llenar un campo solicitado, se deberá mencionar al usuario que lo ingrese.  (E-2) Verificar que el paciente nuevo no se encuentre ya registrado.

## Caso de uso “Baja paciente”

The screenshot shows a window titled "Control de pacientes" with a "Baja" (Discharge) form. The form contains the following fields:

- Numero de paciente: 1100610141
- Nombre (s): STEPHANIE
- Apellido paterno: VARELA
- Apellido materno: PENA
- CURP: STEPH871210JGLDM09
- Fecha de nacimiento: 15 Agosto 1987
- Telefono particular: 56-36-87-89
- Tipo de domicilio: PRIMARIO
- Calle: INGENIEROS, Número exterior: 12P
- Número interior: (empty), Colonia: SATELITE
- Estado: ESTADO DE MÉXICO, Telefono de casa: 98765432
- Municipio: ACAMBAY, Código Postal: 26478
- Fecha de alta: 2011-01-06, Fecha de baja: 9999-12-31

Navigation buttons include "Alta", "Baja", "Modificar", "Eliminar", and "Cancelar". A large blue 3D person icon is also visible on the left side of the form.

Pantalla de baja paciente (P-7)

<b>Caso de uso</b>	Baja paciente
<b>Actores</b>	Usuario, Base de datos
<b>Tipo</b>	Básico
<b>Propósito</b>	Eliminar un paciente registrado.
<b>Resumen</b>	Dar de baja un paciente de la base de datos
<b>Pre condiciones</b>	El paciente debe de estar registrado en la base de datos.
<b>Post condiciones</b>	El paciente aparecerá dado de baja de HCM 2010.
<b>Flujo principal</b>	Se eligió la opción baja en la pantalla (P-6), se elige el paciente y presionamos Eliminar para darlo de baja del HCM 2010.
<b>Subflujo</b>	(S-1) Cancelar. Se regresa a la pantalla (P-6).
<b>Excepciones</b>	(E-1) En caso de que no exista el paciente mostrar el error correspondiente.

## Caso de uso "Modificar paciente"

The screenshot shows a window titled "Control de pacientes" with a "Modificar" section. The form contains the following fields:

- Numero de paciente: 1100610141
- Nombre (s): STEPHANIE
- Apellido paterno: VARELA
- Apellido materno: PENA
- CURP: STEPH871210GLDM09
- Fecha de nacimiento: 15 Agosto 1987
- Telefono particular: 56-36-87-89
- Tipo de domicilio: PRIMARIO
- Calle: INGENIEROS, Número exterior: 12P
- Número interior: , Colonia: SATELITE
- Estado: ESTADO DE MÉXICO, Telefono de casa: 98765432
- Municipio: NAUCALPAN DE JUÁREZ, Código Postal: 26478
- Fecha de alta: 2011-01-06, Fecha de baja: 9999-12-31

Navigation buttons include "Alta", "Baja", "Modificar", "Aceptar", and "Cancelar".

Pantalla de modificar paciente (P-8)

<b>Caso de uso</b>	Modificar paciente
<b>Actores</b>	Usuario, Base de datos
<b>Tipo</b>	Básico
<b>Propósito</b>	Modificar un paciente registrado.
<b>Resumen</b>	Modificar los datos de un paciente.
<b>Pre condiciones</b>	El paciente debe de estar registrado en la base de datos.
<b>Post condiciones</b>	Se almacenaran los cambios realizados en la base de datos.
<b>Flujo principal</b>	Se eligió la opción modificar en la pantalla (P-6), se elige el paciente, se realizan los cambios y presionamos Aceptar para guardar los cambios.
<b>Subflujo</b>	(S-1) Cancelar. Se regresa a la pantalla (P-6).
<b>Excepciones</b>	(E-1) En caso de que no exista el paciente mostrar el error correspondiente.  (E-2) Verificar que los datos modificados sean correctos.

## Caso de uso "Modificar datos de sanidad"

Pantalla de modificar datos de sanidad (P-9)

<b>Caso de uso</b>	Modificar paciente
<b>Actores</b>	Usuario, Base de datos
<b>Tipo</b>	Básico
<b>Propósito</b>	Modificar los datos de sanidad de un paciente registrado
<b>Resumen</b>	Modificar los datos de sanidad un paciente.
<b>Pre condiciones</b>	El paciente debe de estar registrado en la base de datos.
<b>Post condiciones</b>	Se almacenaran los cambios realizados en la base de datos.
<b>Flujo principal</b>	Se eligió la opción datos de sanidad en la pantalla (P-2), se elige el paciente, se realizan los cambios y presionamos Aceptar para guardar los cambios.
<b>Subflujo</b>	(S-1) Cancelar. Se regresa a la pantalla (P-9).
<b>Excepciones</b>	(E-1) En caso de que no exista el paciente mostrar el error correspondiente.  (E-2) Verificar que los datos modificados sean correctos.

## Caso de uso "Asignar credencial"

The screenshot shows a window titled "Credencializacion" with a search icon labeled "Buscar" and an accept icon labeled "Aceptar". The patient information is as follows:

- N° de paciente:** 1100610141
- Nombre:** STEPAHIE VARELA PENA
- Sexo:** FEMENINO **Sangre:** O+
- Nivel socioeconomico:** A/B CLASE RICA
- Vivienda:** CASA BIOCLIMÁTICA O BIOENERGÉTICA
- Grupo Etnico:** NINGUNO
- Religion:** ISLÁMICA
- Enfermedades:** COLITIS ULCERATIVA
- Discapacidades:** CEGUERA
- Alergias:** CHOCOLATE
- Telefono de casa:** 98765432
- Credencial:** (field is empty)

Pantalla de asignar credencial (P-10)

<b>Caso de uso</b>	Asignar credencial
<b>Actores</b>	Usuario, Base de datos, Lector RFID, Tarjeta TISC
<b>Tipo</b>	Básico
<b>Propósito</b>	Asignar una credencial a un paciente
<b>Resumen</b>	Asignar y grabar credencial a un paciente con sus datos de salud.
<b>Pre condiciones</b>	El paciente debe de estar registrado en la base de datos y sus datos de sanidad ya capturados.
<b>Post condiciones</b>	Se guardara el número de la credencial asignada.
<b>Flujo principal</b>	Se eligió la opción datos de credencialización en la pantalla (P-2), se elige el paciente, colocamos la tarjeta sobre el lector y presionamos Aceptar para guardar los datos en la tarjeta
<b>Subflujo</b>	(S-1) Buscar. Busca un nuevo paciente para asignar tarjeta.
<b>Excepciones</b>	(E-1) En caso de que no exista el paciente mostrar el error correspondiente.  (E-2) Verificar que la tarjeta asignada no se haya asignado a otra persona.



## Caso de uso "Recolección"

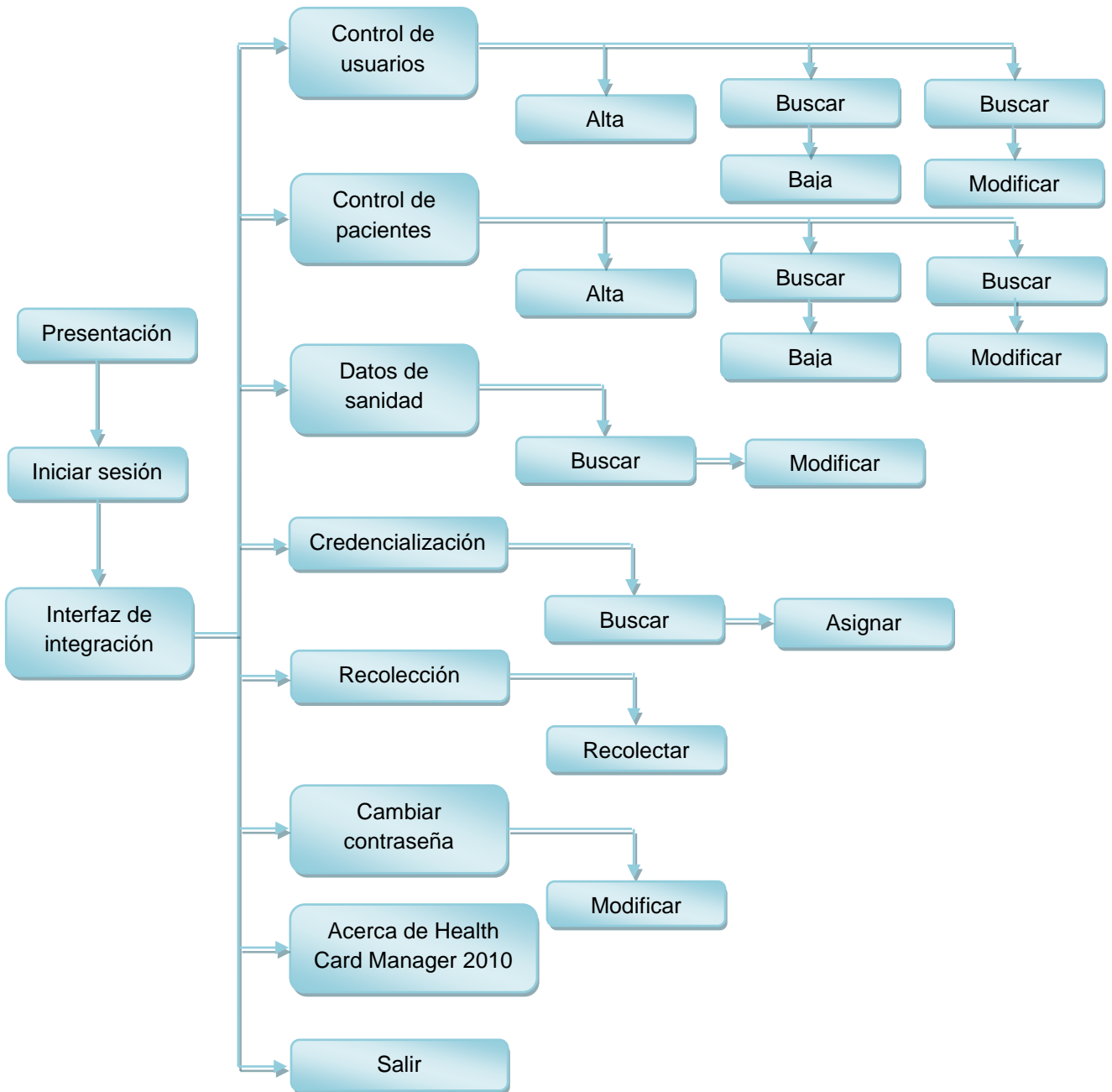
The screenshot shows a software window titled "Recoleccion". On the left side, there is a circular graphic with a green needle and a green checkmark icon labeled "Aceptar". The main area contains a form with the following fields:

- N° de paciente:
- Nombre:
- Sexo:
- Sangre:
- Nivel socioeconomico:
- Vivienda:
- Grupo Etnico:
- Religion:
- Enfermedades:
- Discapacidades:
- Alergias:
- Telefono de casa:
- Credencial:

Pantalla de recolección (P-11)

<b>Caso de uso</b>	Recolección
<b>Actores</b>	Usuario, Lector RFID, Tarjeta TISC
<b>Tipo</b>	Básico
<b>Propósito</b>	Recolectar la información almacenada en la tarjeta
<b>Resumen</b>	Extraer los datos de salud que se almacenaron en la tarjeta.
<b>Pre condiciones</b>	La tarjeta tiene que contener información de salud del paciente
<b>Post condiciones</b>	Sera mostrara la información en la pantalla.
<b>Flujo principal</b>	Se eligió la opción datos de recolección en la pantalla (P-2), se coloca la tarjeta sobre el lector rfid, oprimimos aceptar y es mostrada la información en la pantalla (P-11).
<b>Subflujo</b>	Sin subflujos
<b>Excepciones</b>	(E-1) En caso de no colocar la tarjeta sobre el lector mostrar el error correspondiente.  (E-2) Verificar que la tarjeta a leer sea aceptada por Health Card Manager 2010.

- Mapa de navegación de Health Card Manager 2010.



- Código fuente de Health Card Manager 2010.

“ACR120U.cs”

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Runtime.InteropServices;

namespace Health_Card_Manager
{
    public class ACR120U
    {
        //===== CODIGOS DE ERROR
        =====

        public const int ERR_ACR120_INTERNAL_UNEXPECTED = -1000;
        public const int ERR_ACR120_PORT_INVALID = -2000;
        public const int ERR_ACR120_PORT_OCCUPIED = -2010;
        public const int ERR_ACR120_HANDLE_INVALID = -2020;
        public const int ERR_ACR120_INCORRECT_PARAM = -2030;
        public const int ERR_ACR120_READER_NO_TAG = -3000;
        public const int ERR_ACR120_READER_READ_FAIL_AFTER_OP = -3010;
        public const int ERR_ACR120_READER_NO_VALUE_BLOCK = -3020;
        public const int ERR_ACR120_READER_OP_FAILURE = -3030;
        public const int ERR_ACR120_READER_UNKNOWN = -3040;
        public const int
ERR_ACR120_READER_LOGIN_INVALID_STORED_KEY_FORMAT = -4010;
        public const int
ERR_ACR120_READER_WRITE_READ_AFTER_WRITE_ERROR = -4020;
        public const int ERR_ACR120_READER_DEC_FAILURE_EMPTY = -4030;
        public const int ERR_READER_VALUE_INC_OVERFLOW = -4031;
        public const int ERR_READER_VALUE_OP_FAILURE = -4032;
        public const int ERR_READER_VALUE_INVALID_BLOCK = -4033;
        public const int ERR_READER_VALUE_ACCESS_FAILURE = -4034;
        public const int ERR_READER_NO_RESPONSE = -5000;
    }
}

```

```
//===== PUERTOS USB PARA CONECTAR EL LECTOR
```

```
=====
```

```
public const int ACR120_USB1 = 0;  
public const int ACR120_USB2 = 1;  
public const int ACR120_USB3 = 2;  
public const int ACR120_USB4 = 3;  
public const int ACR120_USB5 = 4;  
public const int ACR120_USB6 = 5;  
public const int ACR120_USB7 = 6;  
public const int ACR120_USB8 = 7;
```

```
//===== TIPOS DE CLAVE PARA TARJETA
```

```
=====
```

```
public const int ACR120_LOGIN_KEYTYPE_A = (int)(0xAA);  
public const int ACR120_LOGIN_KEYTYPE_B = (int)(0xBB);  
public const int ACR120_LOGIN_KEYTYPE_DEFAULT_A = (int)(0xAD);  
public const int ACR120_LOGIN_KEYTYPE_DEFAULT_B = (int)(0xBD);  
public const int ACR120_LOGIN_KEYTYPE_DEFAULT_F = (int)(0xFD);  
public const int ACR120_LOGIN_KEYTYPE_STORED_A = (int)(0xAF);  
public const int ACR120_LOGIN_KEYTYPE_STORED_B = (int)(0xBF);
```

```
[StructLayout(LayoutKind.Sequential)]
```

```
public struct tReaderStatus
```

```
{
```

```
    // 0x01 = Type A; 0x02 = Type B; 0x03 = Type A + Type B
```

```
    public byte MifareInterfaceType;
```

```
    // Bit 0 = Mifare Light; Bit 1 = Mifare1K; Bit 2 = Mifare 4K; Bit 3 = Mifare
```

```
    DESFire
```

```
    // Bit 4 = Mifare UltraLight; Bit 5 = JCOP30; Bit 6 = Shanghai Transport
```

```
    // Bit 7 = MPCOS Combi; Bit 8 = ISO type B, Calypso
```

```
    // Bit 9 - Bit 31 = To be defined
```

```
[MarshalAs(UnmanagedType.ByValArray, SizeConst = 4)]
```

```
    public byte[] CardsSupported;
```

```

// 0x00 = Type A; 0x01 = Type B TAG is being processed
// 0xFF = No TAG is being processed
public byte CardOpMode;

public byte FWI;           // the current FWI value (time out value)
public byte RFU;           // To be defined
public int RFU2;           // To be defined
};

//=====
// INSTRUCCIONES SOPORTADAS POR EL LECTOR ACR120U
//=====

[DllImport("ACR120U.DLL")]
public static extern int ACR120_Open(int ReaderPort);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_Close(int hReader);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_Reset(int hReader);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_Status(int hReader, ref byte pFirmwareVersion,
ref tReaderStatus pReaderStatus);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_ReadRC531Reg(int hReader, byte RegNo, ref
byte pRegData);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_WriteRC531Reg(int hReader, byte RegNo, byte
pRegData);

[DllImport("ACR120U.DLL")]

```

```

    public static extern int ACR120_DirectSend(int hReader, byte DataLength, ref
byte pData,
                                ref byte pResponseDataLength, ref byte
pResponseData, int TimedOut);

    [DllImport("ACR120U.DLL")]
    public static extern int ACR120_DirectReceive(int hReader, byte
RespectedDataLength, ref byte pReceivedDataLength,
                                ref byte pReceivedData, int TimedOut);

    [DllImport("ACR120U.DLL")]
    public static extern int ACR120_RequestDLLVersion(ref byte pVersionInfoLen, ref
byte pVersionInfo);

    [DllImport("ACR120U.DLL")]
    public static extern int ACR120_ReadEEPROM(int hReader, byte RegNo, ref byte
pEEPROMData);

    [DllImport("ACR120U.DLL")]
    public static extern int ACR120_WriteEEPROM(int hReader, byte RegNo, byte
eePROMData);

    [DllImport("ACR120U.DLL")]
    public static extern int ACR120_ReadUserPort(int hReader, ref byte
pUserPortState);

    [DllImport("ACR120U.DLL")]
    public static extern int ACR120_WriteUserPort(int hReader, byte userPortState);

//=====
=====
// INSTRUCCIONES SOPORTADAS POR LA TARJETA

//=====
=====

    [DllImport("ACR120U.DLL")]

```

```

public static extern int ACR120_Select(int hReader, ref byte pResultTagType,
                                       ref byte pResultTagLength, ref byte pResultSN);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_Login(int hReader, byte Sector, int keyType,
                                       byte storedNo, ref byte pKey);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_Read(int hReader, byte Block, ref byte
pBlockData);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_ReadValue(int hReader, byte Block, ref int
pValueData);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_Write(int hReader, byte Block, ref byte
pBlockData);

[DllImport("ACR120U.DLL", CharSet = CharSet.Auto)]
public static extern int ACR120_WriteValue(int hReader, byte Block, int
ValueData);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_WriteMasterKey(int hReader, byte keyNo, ref byte
pKey);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_Inc(int hReader, byte Block,
int value, ref int pNewValue);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_Dec(int hReader, byte Block,
int value, ref int pNewValue);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_Copy(int hReader, byte srcBlock,

```

```

        byte desBlock, ref int pNewValue);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_Power(int hReader, byte State);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_ListTags(int hReader, ref byte pNumTagFound,
ref byte pTagType,
                                ref byte pTagLength, ref byte pSN);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_MultiTagSelect(int hReader, byte TagLength, ref
byte SN,
                                ref byte pResultTagType, ref byte
pResultTagLength,
                                ref byte pResultSN);

[DllImport("ACR120U.DLL")]
public static extern int ACR120_TxDataTelegram(int hReader, byte
SendDataLength, ref byte pSendData,
                                ref byte pReceivedDataLength, ref byte
pReceivedData);
// Error routines
public static string GetErrMsg(int code)
{
    switch (code)
    {
        case -1000:
            return ("-1000: Error inesperado de la biblioteca interna");

        case -2000:
            return ("-2000: Puerto invalido");

        case -2010:
            return ("-2010: Puerto ocupado por otra aplicacion");

        case -2020:

```



```

        return ("-2020: Identificador invalido");

    case -2030:
        return ("-2030: Parametro incorrecto");

    case -3000:
        return ("-3000: No hay tarjeta seleccionada o tarjeta no soportada por la
aplicación");

    case -3010:
        return ("-3010: No se pudo leer después de la operación");

    case -3020:
        return ("-3020: El bloque no contiene valor");

    case -3030:
        return ("-3030: Error en la operación");

    case -3040:
        return ("-3040: Error inesperado del lector ACR120U");

    case -4010:
        return ("-4010: Clave almacenada en formato invalido");

    case -4020:
        return ("-4020: El lector no pudo leer despues de la operación de
escritura");

    case -4030:
        return ("-4030: Decremento invalido(vacio)");

    }
    return (code.ToString() + ": Error inesperado ");
}

public static string GetTagType1(byte XTag)
{

```

```
switch (Convert.ToInt16(XTag))
{
    case 1:
        return ("Mifare Light");

    case 2:
        return ("Mifare 1K");

    case 3:
        return ("Mifare 4K");

    case 4:
        return ("Mifare DESFire");

    case 5:
        return ("Mifare Ultralight");

    case 6:
        return ("JCOP30");

    case 7:
        return ("Shanghai Transport");

    case 8:
        return ("MPCOS Combi");

    case 128:
        return ("ISO Type B, Calypso");
}
return ("Sin identificar por el lector ACR120U");
}
}
```

“cDomicilio.cs”

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Health_Card_Manager
{
    public class cDomicilio
    {
        #region Datos miembro de la clase cDomicilio
        private int idTipo_Domicilio;
        private string calle;
        private string numExt;
        private string numInt;
        private string colonia;
        private int idMunicipio_Delegacion;
        private int idEstado;
        private int codigoPostal;
        private string telefono1;
        #endregion

        #region Metodos Get y Set de la clase cDomicilio
        public int IdTipo_Domicilio
        {
            get { return idTipo_Domicilio;}
            set { idTipo_Domicilio = value; }
        }
        public string Calle
        {
            get { return calle;}
            set { calle = value; }
        }
        public string NumExt
        {
            get { return numExt;}
            set { numExt = value; }
        }
    }
}
```

```

    }
    public string NumInt
    {
        get { return numInt;}
        set { numInt = value; }
    }
    public string Colonia
    {
        get { return colonia;}
        set { colonia = value; }
    }
    public int IdMunicipio_Delegacion
    {
        get { return idMunicipio_Delegacion;}
        set { idMunicipio_Delegacion = value; }
    }
    public int IdEstado
    {
        get { return idEstado; }
        set { idEstado = value; }
    }
    public intCodigoPostal
    {
        get { return codigoPostal;}
        set { codigoPostal = value; }
    }
    public string Telefono1
    {
        get { return telefono1; }
        set { telefono1 = value; }
    }
    #endregion
}
}

```

“conexionMySQL.cs”

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Data;
using MySql.Data.MySqlClient;
using System.Windows.Forms;
using System.Diagnostics;

namespace Health_Card_Manager
{

    public class conexionMySQL
    {

        private MySqlConnection cx = new MySqlConnection("Server=127.0.0.1; User
id=HCM2010; Database=DBHEALTHCARD2010; Password=HCM2010ADMIN;
Connection Timeout=5;");

        public bool conectar()
        {
            try
            {
                if (cx.State == ConnectionState.Open)
                    cx.Close();
                cx.Open();
                return true;
            }
            catch (MySqlException ex)
            {
                DialogResult res= MessageBox.Show(ex.Message, "Error de conexion",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                return false;
            }
        }
    }
}
```

```

public bool desconectar()
{
    try
    {
        if (cx.State == ConnectionState.Open)
            cx.Close();
        cx.Open();
        return true;
    }
    catch (MySqlException ex)
    {
        MessageBox.Show(ex.Message, "Error de desconexion",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}

```

//Selecciona los datos de un catalogo de la base de datos y los regresa en un  
DataReader

```

public MySqlDataReader seleccionarCatalogo(String catalogoDB, String
columnasDB, String condocionDB)
{
    MySqlDataReader dr=null;
    try
    {
        MySqlCommand cmd = new MySqlCommand("SELECT " + columnasDB + "
FROM " + catalogoDB + condocionDB, cx);
        dr = cmd.ExecuteReader();
        return dr;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error de selección",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return dr;
    }
}

```

```

}

//Asigna los datos de un catalogo de la base de datos a un control ComboBox
public void asignarCatalogoLista(String catalogoDB, String columnasDB,
ComboBox nombreLista, Boolean ordenar)
{
    MySqlCommand cmd;
    if (ordenar)
        cmd = new MySqlCommand("SELECT " + columnasDB + " FROM " +
catalogoDB + " ORDER BY " + columnasDB, cx);
    else
        cmd = new MySqlCommand("SELECT " + columnasDB + " FROM " +
catalogoDB, cx);
    MySqlDataReader dr = cmd.ExecuteReader();
    while (dr.Read())
        nombreLista.Items.Add(dr.GetString(0));
    dr.Close();
}

//Obtiene los modulos a los que tiene acceso un usuario
public MySqlDataReader obtenerModulos(int idUsuario)
{
    MySqlDataReader dr = null;
    try
    {
        MySqlCommand cmd = new MySqlCommand("select modulo from
cat_modulos where idmodulo in(select idmodulo from usuarios_modulos where
idusuario=" + idUsuario.ToString() + ")", cx);
        dr = cmd.ExecuteReader();
        return dr;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error de selección",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return dr;
    }
}

```

```

}

//Cambia la contraseña del usuario en la base de datos
public bool cambiarContraseñaDB(int idUsuario, string nuevaContraseña)
{
    MySqlCommand cmd;
    cmd = new MySqlCommand("UPDATE USUARIOS SET CONTRASENA=" +
nuevaContraseña + " WHERE IDUSUARIO=" + idUsuario.ToString(), cx);
    if (cmd.ExecuteNonQuery() == 1)
    {
        frmLogin.uLogeado.Contrasena = nuevaContraseña;
        return true;
    }
    return false;
}

public bool insertarUsuario(string ced, string cur, string nom, string apP,
    string apM, int ide, int ids, int idc, int idr,
    string usu, string con, string fA, string fB, bool act)
{
    try
    {
        if(frmLogin.con.conectar())
        {
            MySqlCommand cmd;
            cmd = new MySqlCommand("INSERT INTO USUARIOS
(CEDULA,CURP,NOMBRE,APELLIDO_PATERNO,"
                +
"APELLIDO_MATERNO,IDESPECIALIDAD,IDSUB_ESPECIALIDAD,IDCLUES,IDROL
,"
                +
"USUARIO,CONTRASENA,FECHA_ALTA,FECHA_BAJA,ACTIVADO) VALUES (" +
ced + "," +
                + cur + "," + nom + "," + apP + "," + apM + "," + ide.ToString() + "," +
ids.ToString()
                + "," + idc.ToString() + ","
                + idr.ToString() + "," + usu + "," + con + "," + fA + ","

```



```

        + fB + "," + act.ToString() + ")", cx);
    if (cmd.ExecuteNonQuery() == 1)
        return true;
    return false;
}
return false;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Error al insertar usuario",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
    return false;
}
}

public bool eliminarUsuario(int idUsuario)
{
    try
    {
        MySqlCommand cmd = new MySqlCommand("DELETE FROM USUARIO
WHERE IDUSUARIO=" + idUsuario.ToString());
        if (cmd.ExecuteNonQuery()==1)
            return true;
        return false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error al borrar usuario",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}

public bool insertarUsuariosModulos(int idUsuario, bool CU, bool CP, bool DS,
bool C, bool R)
{

```

```

//INSERT INTO
USUARIOS_MODULOS(IDUSUARIO,IDMODULO)VALUES(1,2),(1,3)
try
{
    if (frmLogin.con.conectar())
    {
        MySqlCommand cmd;
        if (CU)
            {cmd = new MySqlCommand("INSERT INTO
USUARIOS_MODULOS(IDUSUARIO,IDMODULO)VALUES(" + idUsuario.ToString() +
",2)", cx);
            cmd.ExecuteNonQuery();}
        if (CP)
            {cmd = new MySqlCommand("INSERT INTO
USUARIOS_MODULOS(IDUSUARIO,IDMODULO)VALUES(" + idUsuario.ToString() +
",3)", cx);
            cmd.ExecuteNonQuery();}
        if (DS)
            {cmd = new MySqlCommand("INSERT INTO
USUARIOS_MODULOS(IDUSUARIO,IDMODULO)VALUES(" + idUsuario.ToString() +
",4)", cx);
            cmd.ExecuteNonQuery();}
        if (C)
            {cmd = new MySqlCommand("INSERT INTO
USUARIOS_MODULOS(IDUSUARIO,IDMODULO)VALUES(" + idUsuario.ToString() +
",5)", cx);
            cmd.ExecuteNonQuery();}
        if (R)
            {cmd = new MySqlCommand("INSERT INTO
USUARIOS_MODULOS(IDUSUARIO,IDMODULO)VALUES(" + idUsuario.ToString() +
",6)", cx);
            cmd.ExecuteNonQuery();}
        return true;
    }
    return false;
}
catch (Exception ex)

```

```

    {
        MessageBox.Show(ex.Message, "Error al insertar
modulos", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}

public bool eliminarUsuariosModulos(int idUsuario)
{
    try
    {
        if (frmLogin.con.conectar())
        {
            MySqlCommand cmd;
            cmd = new MySqlCommand("DELETE FROM USUARIOS_MODULOS
WHERE IDUSUARIO=" + idUsuario.ToString(), cx);
            cmd.ExecuteNonQuery();
            return true;
        }
        return false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error al eliminar los módulos anteriores.",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}

public int idCat(string CatalogoDB, string columnaDB, string articulo)
{
    try
    {
        MySqlCommand cmd2;
        cmd2 = new MySqlCommand("SELECT * FROM " + CatalogoDB + "
WHERE " + columnaDB + "=" + articulo + " ", cx);
        MySqlDataReader dr2 = cmd2.ExecuteReader();

```

```

        if (dr2.Read())
        {
            return dr2.GetInt16(0);
        }
        dr2.Close();
        return -1;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Buscar ID" + columnaDB ,
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return -1;
    }
}

public bool bajaUsuario(int idUsuario)
{
    try
    {
        if (frmLogin.con.conectar())
        {
            MySqlCommand cmd;
            cmd = new MySqlCommand("UPDATE USUARIOS SET FECHA_BAJA
            =" + frmLogin.fechaHoyMySQL() + " WHERE IDUSUARIO=" + idUsuario.ToString(),
            cx);

            if (cmd.ExecuteNonQuery() == 1)
                return true;
        }
        return false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error al dar de baja usuario",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}

```

```

    }

    public bool modificarUsuario(int idUsuario, string ced, string cur, string nom, string
apP,
    string apM, int ide, int ids, int idc, int idr,
    string usu, string con, string fA, string fB, bool act)
    {
        try
        {
            if (frmLogin.con.conectar())
            {
                MySqlCommand cmd;
                cmd = new MySqlCommand("UPDATE USUARIOS SET CEDULA=" +
ced + ",CURP="
                + cur + ",NOMBRE=" + nom + ",APELLIDO_PATERNO=" + apP +
",APELLIDO_MATERNO=" + apM + ",IDESPECIALIDAD=" + ide.ToString() +
",IDSUB_ESPECIALIDAD=" + ids.ToString()
                + ", IDCLUES=" + idc.ToString() + ",IDROL="
                + idr.ToString() + ",USUARIO=" + usu + ",CONTRASENA=" + con +
",FECHA_ALTA=" + fA + ",FECHA_BAJA="
                + fB + ",ACTIVADO=" + act.ToString() + " WHERE IDUSUARIO=" +
idUsuario.ToString(), cx);
                if (cmd.ExecuteNonQuery() == 1)
                    return true;
            }
            return false;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Error al modificar usuario",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            return false;
        }
    }

    public MySqlDataReader asignarMunicipios(int idEstado)
    {

```

```

    MySqlConnection dr = null;
    try
    {
        if (frmLogin.con.conectar())
        {
            MySqlCommand cmd;
            cmd = new MySqlCommand("SELECT MUNICIPIO_DELEGACION FROM
CAT_MUNICIPIOS_DELEGACIONES WHERE IDESTADO=" +
idEstado.ToString(),cx);
            dr = cmd.ExecuteReader();
        }
        return dr;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error al asignar municipios.",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return dr;
    }
}

public bool insertarPaciente(string numP, string nom, string apP,
    string apM, string curp, string fN, string fA, string fB, string telPart, string
tipoDom, string calle, string numExt, string numInt,
    string colonia, string municipio, string estado, string CP, string telCasa)
{
    try
    {
        MySqlCommand cmd;
        int idTipoDom,idMunicipio,idEstado;
        idTipoDom = buscarIdElemento("CAT_TIPOS_DOMICILIO",
"IDTIPO_DOMICILIO", " WHERE TIPO_DOMICILIO=" + tipoDom + "");
        idMunicipio = buscarIdElemento("CAT_MUNICIPIOS_DELEGACIONES",
"IDMUNICIPIO_DELEGACION", " WHERE MUNICIPIO_DELEGACION=" + municipio
+ "");
        idEstado = buscarIdElemento("CAT_ESTADOS", "IDESTADO", " WHERE
ESTADO=" + estado + "");

```

```

if (frmLogin.con.conectar())
{
    cmd = new MySqlCommand("INSERT INTO
DOMICILIOS(IDTIPO_DOMICILIO,CALLE,NUMERO_EXT,NUMERO_INT,COLONIA,
IDMUNICIPIO_DELEGACION,IDESTADO," +
        "CODIGO_POSTAL,TELEFONO_1)VALUES(" + idTipoDom + "," +
calle + "," + numExt + "," + numInt + "," + colonia +
        "," + idMunicipio + "," + idEstado + "," + CP + "," + telCasa + ")", cx);
    if (cmd.ExecuteNonQuery() == 1)
    {
        MySqlDataReader dr;
        dr = seleccionarCatalogo("DOMICILIOS", "IDDOMICILIO", " WHERE
IDTIPO_DOMICILIO=" + idTipoDom.ToString() + " AND CALLE=" + calle + " AND " +
            "NUMERO_EXT=" + numExt + " AND NUMERO_INT=" + numInt +
            " AND COLONIA=" + colonia + " AND IDMUNICIPIO_DELEGACION=" +
idMunicipio.ToString() +
            " AND IDESTADO=" + idEstado.ToString() + " AND
CODIGO_POSTAL=" + CP + " AND TELEFONO_1=" + telCasa + "");
        dr.Read();
        int idDom = dr.GetInt32(0);
        dr.Close();
        cmd = new MySqlCommand("INSERT INTO
PACIENTES(NUMERO_PACIENTE,NOMBRE,APELLIDO_PATERNO,APELLIDO_MA
TERNO,CURP," +
"FECHA_NACIMIENTO,IDDOMICILIO,FECHA_ALTA,FECHA_BAJA,TELEFONO_2)
VALUES(" + numP.ToString() + "," + nom + "," + apP + "," + apM + "," +
        curp + "," + fN + "," + idDom.ToString() + "," + fA + "," + fB + "," + telPart
+ ")", cx);
        if (cmd.ExecuteNonQuery() == 1)
            return true;
    }
}
return false;
}
catch (Exception ex)
{

```

```

        MessageBox.Show(ex.Message, "Error al insertar paciente",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}

public bool insertarPaciente(string numP, string nom, string apP,
    string apM, string curp, string fN, int idDom, string fA, string fB, string telPart)
{
    try
    {
        MySqlCommand cmd;
        if (frmLogin.con.conectar())
        {
            cmd = new MySqlCommand("INSERT INTO
PACIENTES(NUMERO_PACIENTE,NOMBRE,APELLIDO_PATERNO,APELLIDO_MA
TERNO,CURP," +
"FECHA_NACIMIENTO,IDDOMICILIO,FECHA_ALTA,FECHA_BAJA,TELEFONO_2)
VALUES(" + numP.ToString() + "," + nom + "," + apP + "," + apM + "," +
    curp + "," + fN + "," + idDom.ToString() + "," + fA + "," + fB + "," + telPart
+ ")", cx);
            if (cmd.ExecuteNonQuery() == 1)
                return true;
        }
        return false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error al insertar paciente",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}

public bool modificarPaciente(string numP, string nom, string apP,
    string apM, string curp, string fN, string fA, string fB, string telPart, string
tipoDom, string calle, string numExt, string numInt,

```



```

string colonia, string municipio, string estado, string CP, string telCasa)
{
    try
    {
        MySqlCommand cmd;
        int idTipoDom, idMunicipio, idEstado;
        idTipoDom = buscarIdElemento("CAT_TIPOS_DOMICILIO",
"IDTIPO_DOMICILIO", " WHERE TIPO_DOMICILIO=" + tipoDom + "");
        idMunicipio = buscarIdElemento("CAT_MUNICIPIOS_DELEGACIONES",
"IDMUNICIPIO_DELEGACION", " WHERE MUNICIPIO_DELEGACION=" + municipio
+ "");
        idEstado = buscarIdElemento("CAT_ESTADOS", "IDESTADO", " WHERE
ESTADO=" + estado + "");
        if (frmLogin.con.conectar())
        {
            cmd = new MySqlCommand("INSERT INTO
DOMICILIOS(IDTIPO_DOMICILIO,CALLE,NUMERO_EXT,NUMERO_INT,COLONIA,I
DMUNICIPIO_DELEGACION,IDESTADO," +
                "CODIGO_POSTAL,TELEFONO_1)VALUES(" + idTipoDom + "," +
calle + "," + numExt + "," + numInt + "," + colonia +
                "," + idMunicipio + "," + idEstado + "," + CP + "," + telCasa + ")", cx);
            if (cmd.ExecuteNonQuery() == 1)
            {
                MySqlDataReader dr;
                dr = seleccionarCatalogo("DOMICILIOS", "IDDOMICILIO", " WHERE
IDTIPO_DOMICILIO=" + idTipoDom.ToString() + " AND CALLE=" + calle + " AND " +
                "NUMERO_EXT=" + numExt + " AND NUMERO_INT=" + numInt +
                " AND COLONIA=" + colonia + " AND IDMUNICIPIO_DELEGACION=" +
idMunicipio.ToString() +
                " AND IDESTADO=" + idEstado.ToString() + " AND
CODIGO_POSTAL=" + CP + " AND TELEFONO_1=" + telCasa + "");
                dr.Read();
                int idDom = dr.GetInt32(0);
                dr.Close();
                cmd = new MySqlCommand("UPDATE PACIENTES SET NOMBRE="
+ nom + ", APELLIDO_PATERNO=" + apP + ", APELLIDO_MATERNO=" + apM + ","
+

```

```

        " CURP=" + curp + ", FECHA_NACIMIENTO=" + fN + ",
IDDOMICILIO=" + idDom.ToString() + ",FECHA_ALTA=" + fA + ",FECHA_BAJA=" +
fB + "," +
        " TELEFONO_2=" + telPart + " WHERE NUMERO_PACIENTE=" +
numP.ToString(), cx);
        if (cmd.ExecuteNonQuery() == 1)
            return true;
    }
}
return false;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Error al actualizar paciente",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    return false;
}
}

```

```

public bool modificarPaciente(string numP, string nom, string apP,
string apM, string curp, string fN, int idDom, string fA, string fB, string telPart)
{
    try
    {
        MySqlCommand cmd;
        if (frmLogin.con.conectar())
        {
            cmd = new MySqlCommand("UPDATE PACIENTES SET NOMBRE=" +
nom + ", APELLIDO_PATERNO=" + apP + ",APELLIDO_MATERNO=" + apM + "," +
" CURP=" + curp + ", FECHA_NACIMIENTO=" + fN + ",IDDOMICILIO="
+ idDom.ToString() + ", FECHA_ALTA=" + fA + "," +
" FECHA_BAJA=" + fB + ",TELEFONO_2=" + telPart + " WHERE
NUMERO_PACIENTE=" + numP.ToString(),cx);
            if (cmd.ExecuteNonQuery() == 1)
                return true;
        }
    }
    return false;
}

```

```

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error al actualizar paciente",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}

public bool modificarPacienteDS(string idP, string nivel, string vivienda,
    string grupo, string religion, string sexo, string sangre)
{
    try
    {
        MySqlCommand cmd;
        int idNivel, idVivienda, idgrupo, idReligion, idSexo, idSangre;
        idNivel =
        frmLogin.con.buscarIdElemento("CAT_NIVELES_SOCIOECONOMICOS",
        "IDNIVEL_SOCIOECONOMICO", " WHERE NIVEL_SOCIOECONOMICO=" + nivel +
        "");
        idVivienda = frmLogin.con.buscarIdElemento("CAT_VIVIENDAS",
        "IDVIVIENDA", " WHERE VIVIENDA=" + vivienda + "");
        idgrupo = frmLogin.con.buscarIdElemento("CAT_GRUPOS_ETNICOS",
        "IDGRUPO_ETNICO", " WHERE GRUPO_ETNICO=" + grupo + "");
        idReligion = frmLogin.con.buscarIdElemento("CAT_RELIGIONES",
        "IDRELIGION", " WHERE RELIGION=" + religion + "");
        idSexo = frmLogin.con.buscarIdElemento("CAT_SEXOS", "IDSEXO", "
        WHERE SEXO=" + sexo + "");
        idSangre = frmLogin.con.buscarIdElemento("CAT_SANGRE", "IDSANGRE",
        " WHERE SANGRE=" + sangre + "");
        if (frmLogin.con.conectar())
        {
            cmd = new MySqlCommand("UPDATE PACIENTES SET
            IDNIVEL_SOCIOECONOMICO=" + idNivel.ToString() + ", IDVIVIENDA=" +
            idVivienda.ToString() + ",IDGRUPO_ETNICO=" + idgrupo.ToString() + ", " +

```

```

        " IDRELIGION=" + idReligion.ToString() + ", IDSEXO=" +
idSexo.ToString() + ",IDSANGRE=" + idSangre.ToString() + " WHERE IDPACIENTE="
+ idP, cx);
        if (cmd.ExecuteNonQuery() == 1)
            return true;
        }
        return false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error al actualizar datos de sanidad",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}

```

//busca el id elemento de un catalogo

```

public int buscarIdElemento(String catalogoDB, String columnaDB, String
condicionDB)
{
    MySqlDataReader dr = null;
    try
    {
        MySqlCommand cmd = new MySqlCommand("SELECT " + columnaDB + "
FROM " + catalogoDB + condicionDB, cx);
        dr = cmd.ExecuteReader();
        dr.Read();
        int i=dr.GetInt32(0);
        dr.Close();
        return i;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error en buscar elemento",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return 0;
    }
}

```

```

}

public bool bajaPaciente(string numPaciente)
{
    try
    {
        if (frmLogin.con.conectar())
        {
            MySqlCommand cmd;
            cmd = new MySqlCommand("UPDATE PACIENTES SET FECHA_BAJA
=" + frmLogin.fechaHoyMySQL() + " WHERE NUMERO_PACIENTE=" +
numPaciente, cx);
            if (cmd.ExecuteNonQuery() == 1)
                return true;
        }
        return false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error al dar de baja usuario",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}

public bool eliminarPacienteEDA(string idPaciente)//Eliminar enfermedades,
discapacidades y alergias
{
    try
    {
        if (frmLogin.con.conectar())
        {
            MySqlCommand cmd;
            cmd = new MySqlCommand("DELETE FROM
PACIENTES_ENFERMEDADES WHERE IDPACIENTE=" + idPaciente, cx);
            cmd.ExecuteNonQuery();

```

```

        cmd = new MySqlCommand("DELETE FROM
PACIENTES_DISCAPACIDADES WHERE IDPACIENTE=" + idPaciente, cx);
        cmd.ExecuteNonQuery();
        cmd = new MySqlCommand("DELETE FROM PACIENTES_ALERGIAS
WHERE IDPACIENTE=" + idPaciente, cx);
        cmd.ExecuteNonQuery();
        return true;
    }
    return false;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Error al eliminar EDA",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    return false;
}
}

public bool ingresarPacienteEDA(string idPaciente, ListBox lstEnf, ListBox lstDis,
ListBox lstAle)//ingresa enfermedades, discapacidades y alergias
{
    try
    {
        if (frmLogin.con.conectar())
        {
            MySqlCommand cmd;
            int i;
            int j = lstEnf.Items.Count;
            if(j>0)
                for(i=0;i<j;i++)
                {
                    cmd = new MySqlCommand("INSERT INTO
PACIENTES_ENFERMEDADES(IDPACIENTE,IDENFERMEDAD) VALUES (" +
idPaciente + "," +
(frmLogin.con.buscarIdElemento("CAT_ENFERMEDADES", "IDENFERMEDAD", "
WHERE ENFERMEDAD=" + lstEnf.Items[i].ToString() + ""))).ToString() + ")", cx);
                    cmd.ExecuteNonQuery();
                }
            }
        }
    }
}

```

```

    }
    j = lstDis.Items.Count;
    if(j>0)
        for (i = 0; i < j ; i++)
        {
            cmd = new MySqlCommand("INSERT INTO
PACIENTES_DISCAPACIDADES(IDPACIENTE,IDDISCAPACIDAD) VALUES (" +
            idPaciente + "," +
(frmLogin.con.buscarIdElemento("CAT_DISCAPACIDADES", "IDDISCAPACIDAD", "
WHERE DISCAPACIDAD=" + lstDis.Items[i].ToString() + """).ToString() + ")", cx);
            cmd.ExecuteNonQuery();
        }
    j = lstAle.Items.Count;
    if (j > 0)
        for (i = 0; i < j; i++)
        {
            cmd = new MySqlCommand("INSERT INTO
PACIENTES_ALERGIAS(IDPACIENTE,IDALERGIA) VALUES (" +
            idPaciente + "," +
(frmLogin.con.buscarIdElemento("CAT_ALERGIAS", "IDALERGIA", " WHERE
ALERGIA=" + lstAle.Items[i].ToString() + """).ToString() + ")", cx);
            cmd.ExecuteNonQuery();
        }
    return true;
}
return false;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Error al eliminar EDA",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    return false;
}
}
public bool leerPacienteEDA(string idPaciente, ListBox lstEnf, ListBox lstDis,
ListBox lstAle)//lee enfermedades, discapacidades y alergias
{

```

```

try
{
    if (frmLogin.con.conectar())
    {
        int i;
        MySqlCommand cmd;
        cmd = new MySqlCommand("SELECT IDENFERMEDAD FROM
PACIENTES_ENFERMEDADES WHERE IDPACIENTE=" + idPaciente, cx);
        MySqlDataReader dr=cmd.ExecuteReader();
        DataTable ids = new DataTable();
        ids.Load(dr);
        dr.Close();
        for (i = 0; i < ids.Rows.Count; i++)

lstEnf.Items.Add(frmLogin.con.buscarElemento("CAT_ENFERMEDADES",
"ENFERMEDAD", " WHERE IDENFERMEDAD=" +
ids.Rows[i].ItemArray[0].ToString()));
        ids = new DataTable();
        cmd = new MySqlCommand("SELECT IDDISCAPACIDAD FROM
PACIENTES_DISCAPACIDADES WHERE IDPACIENTE=" + idPaciente, cx);
        dr = cmd.ExecuteReader();
        ids.Load(dr);
        dr.Close();
        for (i = 0; i < ids.Rows.Count; i++)

lstDis.Items.Add(frmLogin.con.buscarElemento("CAT_DISCAPACIDADES",
"DISCAPACIDAD", " WHERE IDDISCAPACIDAD=" +
ids.Rows[i].ItemArray[0].ToString()));
        ids = new DataTable();
        cmd = new MySqlCommand("SELECT IDALERGIA FROM
PACIENTES_ALERGIAS WHERE IDPACIENTE=" + idPaciente, cx);
        dr = cmd.ExecuteReader();
        ids.Load(dr);
        dr.Close();
        for (i = 0; i < ids.Rows.Count; i++)
            lstAle.Items.Add(frmLogin.con.buscarElemento("CAT_ALERGIAS",
"ALERGIA", " WHERE IDALERGIA=" + ids.Rows[i].ItemArray[0].ToString()));

```



```

        ids = new DataTable();
        return true;
    }
    return false;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Error al eliminar EDA",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
    return false;
}
}

public string buscarElemento(String catalogoDB, String columnaDB, String
condocionDB)
{
    MySqlDataReader dr = null;
    try
    {
        MySqlCommand cmd = new MySqlCommand("SELECT " + columnaDB + "
FROM " + catalogoDB + condocionDB, cx);
        dr = cmd.ExecuteReader();
        string ele="";
        if(dr.Read())
            ele = dr.GetString(0);
        dr.Close();
        return ele;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error en buscar elemento",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
        return "";
    }
}

public void insertarCredencial(string idPaciente,String numCredencial)

```

```

{
    try
    {
        if (frmLogin.con.conectar())
        {
            MySqlCommand cmd;
            cmd = new MySqlCommand("UPDATE CREDENCIALES SET
FECHA_BAJA=" + frmLogin.fechaHoyMySQL() + " WHERE IDPACIENTE=" +
idPaciente.ToString(), cx);
            cmd.ExecuteNonQuery();
            cmd = new MySqlCommand("INSERT INTO
CREDENCIALES(IDPACIENTE,NUMERO_CREDENCIAL,FECHA_ALTA,FECHA_BAJ
A) VALUES (" + idPaciente + "," + numCredencial + "," + frmLogin.fechaHoyMySQL()
+ "','9999/12/31')", cx);
            cmd.ExecuteNonQuery();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error al insertar
credencial", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

public bool insertarAuditoria(string idUsu, string idMod, string idMov,
string des, string fec, string hor)
{
    try
    {
        MySqlCommand cmd;
        if (frmLogin.con.conectar())
        {
            cmd = new MySqlCommand("INSERT INTO
AUDITORIA(IDUSUARIO,IDMODULO,IDMOVIMIENTO,DESCRIPCION,FECHA," +
"HORA) VALUES(" + idUsu + "," + idMod + "," + idMov + "," + des + "," +
fec + "," + hor + ")", cx);

```

```
        if (cmd.ExecuteNonQuery() == 1)
            return true;
    }
    return false;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Error al insertar auditoria",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
    return false;
}
}
}
}
```

“cPaciente.cs”

```
using System;
using System.Collections.Generic;
using System.Text;
using MySql.Data.MySqlClient;

namespace Health_Card_Manager
{

    public class cPaciente : cDomicilio
    {
        #region Datos miembro de la clase cUsuario
        private int idPaciente;
        private int numPaciente;
        private string nombre;
        private string apPaterno;
        private string apMaterno;
        private string curp;
        private string fechaNacimiento;
        private int idNivelSocioeconomico;
        private int idGrupoEtnico;
        private int idReligion;
        private int idSexo;
        private int idSangre;
        private int idDomicilio;
        private string fechaAlta;
        private string fechaBaja;
        private string telefono2;
        #endregion

        #region Constructores de la clase Paciente
        public cPaciente()
        {
            idPaciente=0;
            numPaciente=0;
            nombre=string.Empty;
        }
    }
}
```

```

apPaterno=string.Empty;
apMaterno=string.Empty;
curp=string.Empty;
fechaNacimiento=string.Empty;
idNivelSocioeconomico=0;
idGrupoEtnico=0;
idReligion=0;
idSexo=0;
idSangre = 0;
idDomicilio=0;
fechaAlta=string.Empty;
fechaBaja=string.Empty;
telefono2=string.Empty;
}
#endregion

```

#region Metodos Get y Set de la clase cUsuario

```

public int IdPaciente
{
    get { return idPaciente; }
    set { idPaciente = value; }
}
public int NumPaciente
{
    get { return numPaciente; }
    set { numPaciente = value; }
}
public string Nombre
{
    get { return nombre; }
    set { nombre = value; }
}
public string ApPaterno
{
    get { return apPaterno; }
    set { apPaterno = value; }
}
}

```

```
public string ApMaterno
{
    get { return apPaterno; }
    set { apPaterno = value; }
}
public string Curp
{
    get { return curp; }
    set { curp = value; }
}
public string FechaNacimiento
{
    get { return fechaNacimiento; }
    set { fechaNacimiento = value; }
}
public int IdNivelSocioeconomico
{
    get { return idNivelSocioeconomico; }
    set { idNivelSocioeconomico = value; }
}
public int IdGrupoEtnico
{
    get { return idGrupoEtnico; }
    set { idGrupoEtnico = value; }
}
public int IdReligion
{
    get { return idReligion; }
    set { idReligion = value; }
}
public int IdSexo
{
    get { return idSexo; }
    set { idSexo = value; }
}
public int IdSangre
{
```

```

    get { return idSangre; }
    set { idSangre = value; }
}
public int IdDomicilio
{
    get { return idDomicilio; }
    set { idDomicilio = value; }
}
public string FechaAlta
{
    get { return fechaAlta; }
    set { fechaAlta = value; }
}
public string FechaBaja
{
    get { return fechaBaja; }
    set { fechaBaja = value; }
}
public string Telefono2
{
    get { return telefono2; }
    set { telefono2 = value; }
}
#endregion

```

**#region** Metodos de la clase paciente

```

public void BDpacienteTOcPaciente(MySqlDataReader dr)
{
    IdPaciente = dr.GetInt32("IDPACIENTE");
    NumPaciente = dr.GetInt32("NUMERO_PACIENTE");
    Nombre = dr.GetString("NOMBRE");
    ApPaterno = dr.GetString("APELLIDO_PATERNO");
    ApMaterno = dr.GetString("APELLIDO_MATERNO");
    Curp = dr.GetString("CURP");
    FechaNacimiento = dr.GetString("FECHA_NACIMIENTO");
    IdNivelSocioeconomico = dr.GetInt32("IDNIVEL_SOCIOECONOMICO");
}

```

```
IdGrupoEtnico = dr.GetInt32("IDGRUPO_ETNICO");
IdReligion = dr.GetInt32("IDRELIGION");
IdSexo = dr.GetInt32("IDSEXO");
IdSangre = dr.GetInt32("IDSANGER");
IdDomicilio = dr.GetInt32("IDDOMICILIO");
FechaAlta = dr.GetString("FECHA_ALTA");
FechaBaja = dr.GetString("FECHA_BAJA");
Telefono2 = dr.GetString("TELEFONO_2");
}
#endregion
}
}
```



"cUsuario.cs"

```
using System;
using System.Collections.Generic;
using System.Text;
using MySql.Data.MySqlClient;

namespace Health_Card_Manager
{
    public class cUsuario
    {
        #region Datos miembro de la clase cUsuario
        private int idUsuario;
        private string cedula;
        private string curp;
        private string nombre;
        private string apPaterno;
        private string apMaterno;
        private int idEspecialidad;
        private int idsubEspecialidad;
        private int idClues;
        private int idRol;
        private string usuario;
        private string contrasena;
        private string fechaAlta; //String tempDate = tDate.ToString("yyyy/MM/dd");
        private string fechaBaja;
        private bool activado;
        #endregion

        #region Constructores de la clase Usuario
        public cUsuario()
        {
            idUsuario=0;
            cedula = null;
            curp = string.Empty;
            nombre = string.Empty;
            apPaterno = string.Empty;
        }
    }
}
```

```

apMaterno = string.Empty;
idEspecialidad = 0;
idsubEspecialidad = 0;
idClues =0;
idRol = 0;
usuario = string.Empty;
contrasena = string.Empty;
fechaAlta = DateTime.Now.TimeOfDay.ToString();
fechaBaja = "9999/12/31";
activado=true;
}

```

```

public cUsuario(int idUs, string ce, string cu,string no, string apPa,
    string apMa, int idEs, int idsu, int idDo, int idCl, int idRo,
    string us, string co, string fa, string fb, bool ac)

```

```

{
idUsuario=idUs;
cedula = ce;
curp = cu;
nombre = no;
apPaterno = apPa;
apMaterno = apMa;
idEspecialidad = idEs;
idsubEspecialidad = idsu;
idClues =idCl;
idRol = idRo;
usuario = us;
contrasena = co;
fechaAlta = fa;
fechaBaja = fb;
activado=ac;
}

```

`#endregion`

`#region` Metodos Get y Set de la clase cUsuario

```

public int IdUsuario
{

```

```

    get { return idUsuario;}
    set { idUsuario = value; }
}
public string Cedula
{
    get { return cedula;}
    set { cedula = value; }
}
public string Curp
{
    get { return curp;}
    set { curp = value; }
}
public string Nombre
{
    get { return nombre;}
    set { nombre = value; }
}
public string ApPaterno
{
    get { return apPaterno;}
    set { apPaterno = value; }
}
public string ApMaterno
{
    get { return apMaterno;}
    set { apMaterno = value; }
}
public int IdEspecialidad
{
    get { return idEspecialidad;}
    set { idEspecialidad = value; }
}
public int IdsubEspecialidad
{
    get { return idsubEspecialidad;}
    set { idsubEspecialidad = value; }
}

```

```
}  
public int IdClues  
{  
    get { return idClues;}  
    set { idClues = value; }  
}  
public int IdRol  
{  
    get { return idRol;}  
    set { idRol = value; }  
}  
public string Usuario  
{  
    get { return usuario;}  
    set { usuario = value; }  
}  
public string Contraseña  
{  
    get { return contraseña;}  
    set { contraseña = value; }  
}  
public string FechaAlta  
{  
    get { return fechaAlta;}  
    set { fechaAlta = value; }  
}  
public string FechaBaja  
{  
    get { return fechaBaja;}  
    set { fechaBaja = value; }  
}  
public bool Activado  
{  
    get { return activado;}  
    set { activado = value; }  
}  
#endregion
```

```

#region Metodos de la clase cUsuario
public string getNombreCompleto()
{
    return Nombre + " " + ApPaterno + " " + ApMaterno;
}

public void BDusuarioTOcUsuario(MySqlDataReader dr)
{
    IdUsuario = dr.GetInt32("IDUSUARIO");
    Cedula = dr.GetString("CEDULA");
    Curp = dr.GetString("CURP");
    Nombre = dr.GetString("NOMBRE");
    ApPaterno = dr.GetString("APELLIDO_PATERNO");
    ApMaterno = dr.GetString("APELLIDO_MATERNO");
    IdEspecialidad = dr.GetInt32("IDESPECIALIDAD");
    IdsubEspecialidad = dr.GetInt32("IDSUB_ESPECIALIDAD");
    IdClues = dr.GetInt32("IDCLUES");
    IdRol = dr.GetInt32("IDROL");
    Usuario = dr.GetString("USUARIO");
    Contraseña = dr.GetString("CONTRASENA");
    FechaAlta = dr.GetString("FECHA_ALTA");
    FechaBaja = dr.GetString("FECHA_BAJA");
    Activado = dr.GetBoolean("ACTIVADO");
}
#endregion
}
}

```

“frmBuscarPaciente.cs”

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Health_Card_Manager
{
    public partial class frmBuscarPaciente : Form
    {
        public frmBuscarPaciente()
        {
            InitializeComponent();
        }

        private void btnBus_Click(object sender, EventArgs e)
        {
            obtenerPacientes();
        }

        public void obtenerPacientes()
        {
            try
            {
                dgPac.Visible = true;
                dgPac.Rows.Clear();
                dgPac.Rows[0].Cells[0].Value = "";
                if (frmLogin.con.conectar())
                {
                    string cmdStr;
```

```

if (!frmControlPacientes.incluirbajas)
    cmdStr = " WHERE FECHA_BAJA > " + frmLogin.fechaHoyMySQL() +
"";

else
    cmdStr = " ";
if (txtBus.Text != "")
    cmdStr += " AND NUMERO_PACIENTE=" + txtBus.Text + "";

MySqlDataReader dr = frmLogin.con.seleccionarCatalogo("PACIENTES",
"IDPACIENTE", cmdStr);
DataTable t = new DataTable();
t.Load(dr);
int h = t.Rows.Count;
dr = frmLogin.con.seleccionarCatalogo("PACIENTES",
"IDPACIENTE,IDDOMICILIO,NUMERO_PACIENTE,NOMBRE,APELLIDO_PATERO,
APELLIDO_MATERNO", cmdStr + " ORDER BY(APELLIDO_PATERO)");
int i = 0;
while (dr.Read())
{
    if (i < h - 1)
        dgPac.Rows.Add();
    dgPac.Rows[i].Cells[0].Value = dr.GetString("NUMERO_PACIENTE");
    dgPac.Rows[i].Cells[1].Value = dr.GetString("APELLIDO_PATERO") +
" " + dr.GetString("APELLIDO_MATERNO") + " " + dr.GetString("NOMBRE");
    dgPac.Rows[i].Cells[2].Value = dr.GetString("IDPACIENTE");
    dgPac.Rows[i].Cells[3].Value = dr.GetString("IDDOMICILIO");
    this.Refresh();
    i += 1;
}
dr.Close();
dgPac.Rows[0].Selected = true;
if (dgPac.RowCount == 1 && dgPac.Rows[0].Cells[0].Value.ToString() ==
"")
{
    if (txtBus.Text == "")
        MessageBox.Show("No existen pacientes o estan dados de baja.",
"Buscar paciente", MessageBoxButtons.OK, MessageBoxIcon.Warning);
}

```

```

        else
            MessageBox.Show("No existe el paciente " + txtBus.Text + " o esta
            dado de baja..", "Buscar paciente", MessageBoxButtons.OK,
            MessageBoxIcon.Warning);
            btnAce.Visible = false;
            return;
        }
        btnAce.Visible = true;
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Obtener pacientes",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

private void txtBus_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == Convert.ToChar(Keys.Return))
        obtenerPacientes();
}

private void btnAce_Click(object sender, EventArgs e)
{
    if (dgPac.RowCount == 1)
    {
        frmControlPacientes.pBuscado.IdPaciente =
        int.Parse(dgPac.Rows[0].Cells[2].Value.ToString());
        frmControlPacientes.pBuscado.IdDomicilio =
        int.Parse(dgPac.Rows[0].Cells[3].Value.ToString());
    }
    else
    {
        frmControlPacientes.pBuscado.IdPaciente =
        int.Parse(dgPac.CurrentRow.Cells[2].Value.ToString());
    }
}

```



```
        frmControlPacientes.pBuscado.IdDomicilio =  
int.Parse(dgPac.CurrentRow.Cells[3].Value.ToString());  
    }  
    this.Close();  
}  
  
}  
}
```

"frmBuscarPacienteS.cs"

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Health_Card_Manager
{
    public partial class frmBuscarPacienteS : Form
    {
        private bool ds;
        public frmBuscarPacienteS(bool frm)
        {
            InitializeComponent();
            this.ds = frm;
        }

        private void btnBus_Click(object sender, EventArgs e)
        { obtenerPacientesS();}

        public void obtenerPacientesS()
        {
            try
            {
                dgPacS.Visible = true;
                dgPacS.Rows.Clear();
                dgPacS.Rows[0].Cells[0].Value = "";
                if (frmLogin.con.conectar())
                {
                    string cmdStr;
                    cmdStr = " WHERE FECHA_BAJA > " + frmLogin.fechaHoyMySQL() + "";
                    if (txtBus.Text != "")
```

```

        cmdStr += " AND NUMERO_PACIENTE=" + txtBus.Text + "";
        MySqlDataReader dr = frmLogin.con.seleccionarCatalogo("PACIENTES",
"IDPACIENTE", cmdStr);
        DataTable t = new DataTable();
        t.Load(dr);
        int h = t.Rows.Count;
        dr = frmLogin.con.seleccionarCatalogo("PACIENTES",
"IDPACIENTE,NUMERO_PACIENTE,NOMBRE,APELLIDO_PATERNO,APELLIDO_M
ATERNO", cmdStr + " ORDER BY(APELLIDO_PATERNO)");
        int i = 0;
        while (dr.Read())
        {
            if (i < h - 1)
                dgPacS.Rows.Add();
            dgPacS.Rows[i].Cells[0].Value = dr.GetString("NUMERO_PACIENTE");
            dgPacS.Rows[i].Cells[1].Value = dr.GetString("APELLIDO_PATERNO")
+ " " + dr.GetString("APELLIDO_MATERNO") + " " + dr.GetString("NOMBRE");
            dgPacS.Rows[i].Cells[2].Value = dr.GetString("IDPACIENTE");
            this.Refresh();
            i += 1;
        }
        dr.Close();
        dgPacS.Rows[0].Selected = true;
        if (dgPacS.RowCount == 1 && dgPacS.Rows[0].Cells[0].Value.ToString()
== "")
        {
            if (txtBus.Text == "")
                MessageBox.Show("No existen pacientes o estan dados de baja.",
"Buscar paciente", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            else
                MessageBox.Show("No existe el paciente " + txtBus.Text + " o esta
dado de baja..", "Buscar paciente", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
            btnAce.Visible = false;
            return;
        }
        btnAce.Visible = true;

```

```

    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Obtener pacientes",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

private void txtBus_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == Convert.ToChar(Keys.Return))
        obtenerPacientesS();
}

private void btnAce_Click(object sender, EventArgs e)
{
    if(ds)
    {
        if (dgPacS.RowCount == 1)
        {
            frmDatosSanidad.NP =
int.Parse(dgPacS.Rows[0].Cells[0].Value.ToString());
            frmDatosSanidad.idP =
int.Parse(dgPacS.Rows[0].Cells[2].Value.ToString());
        }
        else
        {
            frmDatosSanidad.NP =
int.Parse(dgPacS.CurrentRow.Cells[0].Value.ToString());
            frmDatosSanidad.idP =
int.Parse(dgPacS.CurrentRow.Cells[2].Value.ToString());
        }
    }
    else
    {
        if (dgPacS.RowCount == 1)

```

```
    {  
        frmCredencializacion.NP =  
int.Parse(dgPacS.Rows[0].Cells[0].Value.ToString());  
        frmCredencializacion.idP =  
int.Parse(dgPacS.Rows[0].Cells[2].Value.ToString());  
    }  
    else  
    {  
        frmCredencializacion.NP =  
int.Parse(dgPacS.CurrentRow.Cells[0].Value.ToString());  
        frmCredencializacion.idP =  
int.Parse(dgPacS.CurrentRow.Cells[2].Value.ToString());  
    }  
    }  
    this.Close();  
    }  
    }  
}
```

"frmBuscarUsuario.cs"

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Health_Card_Manager
{
    public partial class frmBuscarUsuario : Form
    {
        public frmBuscarUsuario()
        {
            InitializeComponent();
        }

        private void btnBus_Click(object sender, EventArgs e)
        {
            obtenerUsuarios();
        }

        public void obtenerUsuarios()
        {
            try
            {
                dgUsu.Visible = true;
                dgUsu.Rows.Clear();
                dgUsu.Rows[0].Cells[0].Value = "";
                if (frmLogin.con.conectar())
                {
                    string cmdStr;
                    if (!frmControlUsuarios.incluirBajas)
```

```

        cmdStr= " WHERE USUARIO<> " + frmLogin.uLogeado.Usuario + "
AND FECHA_BAJA > " + frmLogin.fechaHoyMySQL() + " ORDER
BY(APELLIDO_PATERNO)";
    else
        cmdStr = " WHERE USUARIO<> " + frmLogin.uLogeado.Usuario + "
ORDER BY(APELLIDO_PATERNO)";
        if (txtBus.Text != "")
            cmdStr += "AND USUARIO=" + txtBus.Text + """;
        MySqlDataReader dr = frmLogin.con.seleccionarCatalogo("USUARIOS",
"IDUSUARIO", cmdStr);
        DataTable t = new DataTable();
        t.Load(dr);
        int h = t.Rows.Count;
        dr = frmLogin.con.seleccionarCatalogo("USUARIOS",
"IDUSUARIO,USUARIO,NOMBRE,APELLIDO_PATERNO,APELLIDO_MATERNO",
cmdStr);

        int i = 0;
        while (dr.Read())
        {
            if (i < h - 1)
                dgUsu.Rows.Add();
            dgUsu.Rows[i].Cells[0].Value = dr.GetString("USUARIO");
            dgUsu.Rows[i].Cells[1].Value = dr.GetString("APELLIDO_PATERNO") +
" " + dr.GetString("APELLIDO_MATERNO") + " " + dr.GetString("NOMBRE");
            this.Refresh();
            i += 1;
        }
        dr.Close();
        dgUsu.Rows[0].Selected = true;
        if (dgUsu.RowCount == 1 && dgUsu.Rows[0].Cells[0].Value.ToString() ==
"")
        {
            if(txtBus.Text=="")
                MessageBox.Show("No existen usuarios o estan dados de baja.",
"Buscar usuario", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            else

```

```

        MessageBox.Show("No existe el usuario " + txtBus.Text + " o esta
        dado de baja..", "Buscar usuario", MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
        btnAce.Visible = false;
        return;
    }
    btnAce.Visible = true;
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Obtener
    usuarios.", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

private void txtBus_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == Convert.ToChar(Keys.Return))
        obtenerUsuarios();
}

private void button1_Click(object sender, EventArgs e)
{
    if (dgUsu.RowCount == 1)
        frmControlUsuarios.uBuscado.Usuario =
        dgUsu.Rows[0].Cells[0].Value.ToString();
    else
        frmControlUsuarios.uBuscado.Usuario =
        dgUsu.CurrentRow.Cells[0].Value.ToString();
    this.Close();
}
}
}
}

```



“frmContrasena.cs”

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Health_Card_Manager
{
    public partial class frmContrasena : Form
    {
        public frmContrasena()
        {
            InitializeComponent();
            gpoCon.Text = frmLogin.uLogeado.Usuario;
        }

        private void btnAce_Click(object sender, EventArgs e)
        {
            cambiarContrasena();
        }

        public void cambiarContrasena()
        {
            if (validarCamposGrupo(gpoCon))
                if (txtNue1.Text.CompareTo(txtNue2.Text) == 0)
                {
                    if (txtAnt.Text.CompareTo(frmLogin.uLogeado.Contrasena) == 0)
                    {
                        if (frmLogin.con.conectar())
                            if (frmLogin.con.cambiarContrasenaDB(frmLogin.uLogeado.IdUsuario, txtNue1.Text))
                                {
                                    this.Close();
                                }
                    }
                }
        }
    }
}
```

```

        MessageBox.Show("La contraseña fue cambiada exitosamente.",
"Cambiar contraseña", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
        MessageBox.Show("La contraseña no puede ser cambiada.",
"Cambiar contraseña", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
        MessageBox.Show("La contraseña anterior con coincide.", "Cambiar
contraseña", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
        MessageBox.Show("La confirmacion de la nueva contraseña no
coincide.", "Cambiar contraseña", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

private bool validarCamposGrupo(GroupBox grupo)
{
    foreach (Control c in grupo.Controls)
        if (c is TextBox && c.Text == "")
            {
                MessageBox.Show("Ingresar el campo " + c.Tag + ". ", "Iniciar Sesión",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return false;
            }
    return true;
}

private void txtNue2_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == Convert.ToChar(Keys.Return))
        cambiarContrasena();
}

private void btnCan_Click(object sender, EventArgs e)
{this.Dispose();}
}
}

```

"frmControlPacientes.cs"

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using System.Globalization;

namespace Health_Card_Manager
{
    public partial class frmControlPacientes : Form
    {
        public static cPaciente pBuscado = new cPaciente();
        public static bool incluirbajas = true;
        public frmControlPacientes()
        {
            InitializeComponent();
            inicializarControles();
        }

        private void cmbEst_SelectedIndexChanged(object sender, EventArgs e)
        {
            cmbMun.Items.Clear();
            MySqlDataReader dr = frmLogin.con.asignarMunicipios(cmbEst.SelectedIndex
+ 1);
            while (dr.Read())
                cmbMun.Items.Add(dr.GetString("MUNICIPIO_DELEGACION"));
            if(cmbMun.Items.Count>0)
                cmbMun.SelectedIndex=0;
        }
    }
}
```

```

private void btnAlt_Click(object sender, EventArgs e)
{
    gpoDatos.Text = "A l t a";
    gpoDatos.Enabled = true;
    gpoDireccion.Enabled = true;
    btnAlt.Enabled = false;
    btnBaj.Enabled = false;
    btnCam.Enabled = false;
    btnAE.Enabled = true;
    btnCan.Enabled = true;
    btnAE.BackgroundImage=ilAE.Images[0];
    lblAE.Text="Aceptar";
    Refresh();
    elegirElementos();//elige los index 0 de los combobox

    //frmControlUsuarios

    txtFal.Text = frmLogin.fechaHoyMySQL();
    txtFba.Text = "9999-12-31";
    txtNP.Text=generarNumEmp();
}

public string generarNumEmp()
{
    string h = "00" + DateTime.Now.Hour.ToString();
    string m = "00" + DateTime.Now.Minute.ToString();
    string s = "00" + DateTime.Now.Second.ToString();
    string dia = "00" + DateTime.Now.DayOfYear.ToString();
    return DateTime.Now.Year.ToString().Substring(2, 2) +
    dia.Substring(dia.Length - 3, 3) + h.Substring(h.Length - 2, 1) + m.Substring(m.Length -
    2, 2) + s.Substring(s.Length - 2, 2);
}

public void elegirElementos();//elegimos el primer elemento de cada lista

```

```
{  
    if (cmbAnho.Items.Count > 0)  
    {  
        cmbAnho.SelectedIndex = 0;  
        cmbAnho.Text = cmbAnho.SelectedItem.ToString();  
    }  
  
    if (cmbMes.Items.Count > 0)  
    {  
        cmbMes.SelectedIndex = 0;  
        cmbMes.Text = cmbMes.SelectedItem.ToString();  
    }  
  
    if (cmbDia.Items.Count > 0)  
    {  
        cmbDia.SelectedIndex = 0;  
        cmbDia.Text = cmbDia.SelectedItem.ToString();  
    }  
  
    if (cmbTD.Items.Count > 0)  
    {  
        cmbTD.SelectedIndex = 0;  
        cmbTD.Text = cmbTD.SelectedItem.ToString();  
    }  
  
    if (cmbMun.Items.Count > 0)  
    {  
        cmbMun.SelectedIndex = 0;  
        cmbMun.Text = cmbMun.SelectedItem.ToString();  
    }  
  
    if (cmbEst.Items.Count > 0)  
    {  
        cmbEst.SelectedIndex = 0;  
        cmbEst.Text = cmbEst.SelectedItem.ToString();  
    }  
}
```

```

public void inicializarControles()
{
    for (int i = 1; i < 32; i++)
    {
        cmbDia.Items.Add(i);
    }

    //llenamos combobox con meses
    CultureInfo cultura = new CultureInfo("es-ES", false);
    foreach (string mx in cultura.DateTimeFormat.MonthNames)
    {
        if (mx.CompareTo("") != 0)
            cmbMes.Items.Add(cultura.TextInfo.ToTitleCase(mx));
    }

    //llenamos combobox con anhos
    for (int i = 1910; i < DateTime.Today.Year + 1; i++)
    {
        cmbAnho.Items.Add(i);
    }
    if (frmLogin.con.conectar())
    {
        frmLogin.con.asignarCatalogoLista("CAT_ESTADOS", "ESTADO", cmbEst,
false);
        frmLogin.con.asignarCatalogoLista("CAT_MUNICIPIOS_DELEGACIONES",
"MUNICIPIO_DELEGACION", cmbMun, false);
        frmLogin.con.asignarCatalogoLista("CAT_TIPOS_DOMICILIO",
"TIPO_DOMICILIO", cmbTD, false);
    }
}

private void txtNom_KeyPress(object sender, KeyPressEventArgs e)
{frmControlUsuarios.evitarTeclas(e, true, false, false);}

private void txtApp_KeyPress(object sender, KeyPressEventArgs e)

```

```

{frmControlUsuarios.evitarTeclas(e, true, false, false);}

private void txtCur_KeyPress(object sender, KeyPressEventArgs e)
{frmControlUsuarios.evitarTeclas(e,false, false, true);}

private void textBox4_KeyPress(object sender, KeyPressEventArgs e)
{ frmControlUsuarios.evitarTeclas(e, false, true, true); }

private void cmbMes_TextUpdate(object sender, EventArgs e)
{if(cmbMes.Items.Count>0)
    cmbMes.SelectedIndex = 0;}

private void cmbDia_TextUpdate(object sender, EventArgs e)
{
    if (cmbDia.Items.Count > 0)
        cmbDia.SelectedIndex = 0;
}

private void cmbAnho_TextUpdate(object sender, EventArgs e)
{
    if (cmbAnho.Items.Count > 0)
        cmbAnho.SelectedIndex = 0;
}

private void cmbEst_TextUpdate(object sender, EventArgs e)
{
    if (cmbEst.Items.Count > 0)
        cmbEst.SelectedIndex = 0;
}

private void cmbMun_TextUpdate(object sender, EventArgs e)
{
    if (cmbMun.Items.Count > 0)
        cmbMun.SelectedIndex = 0;
}

```

```

private void cmbTD_TextUpdate(object sender, EventArgs e)
{
    if (cmbTD.Items.Count > 0)
        cmbTD.SelectedIndex = 0;
}

private void btnBus_Click(object sender, EventArgs e)
{
    if (gpoDireccion.Enabled)
        if (gpoDatos.Text.CompareTo("A l t a") == 0)
            buscarPaciente();
        else
            buscarDireccion();
    else
        gpoDireccion.Enabled = true;
}

public void buscarPaciente()
{
    if (gpoDatos.Text.CompareTo("A l t a") == 0 )
        buscarDireccion();

    else if (gpoDatos.Text.CompareTo("B a j a") == 0 ||
gpoDatos.Text.CompareTo("M o d i f i c a r") == 0)
    {
        pBuscado.IdPaciente = 0;
        pBuscado.IdDomicilio = 0;
        incluirbajas = false;
        if (gpoDatos.Text.CompareTo("M o d i f i c a r") == 0)//
            incluirbajas = true;
        frmBuscarPaciente a = new frmBuscarPaciente();
        a.ShowDialog();
        if (pBuscado.IdPaciente > 0)
        {
            MySqlDataReader dr = frmLogin.con.seleccionarCatalogo("PACIENTES",
            "*", " WHERE IDPACIENTE=" + pBuscado.IdPaciente.ToString());

```



```

if (dr.Read())
{
    if (gpoDatos.Text.CompareTo("B a j a") == 0)
        gpoDatos.Enabled = false;
    else
    {
        gpoDatos.Enabled = true;
        gpoDireccion.Enabled = false;
    }
    txtNP.Text = dr.GetString("NUMERO_PACIENTE");
    txtNom.Text = dr.GetString("NOMBRE");
    txtApp.Text = dr.GetString("APELLIDO_PATERNO");
    txtApm.Text = dr.GetString("APELLIDO_MATERNO");
    txtCur.Text = dr.GetString("CURP");
    string[] f =
frmLogin.convetirFechaMySQL(dr.GetString("FECHA_NACIMIENTO")).Split('-');
    cmbAnho.Text = f[0];
    int i = int.Parse(f[1]) - 1;
    if (cmbMes.Items.Count > 0)
    {
        cmbMes.SelectedIndex = i;
        cmbMes.Text = cmbMes.SelectedItem.ToString();
    }
    Refresh();
    cmbDia.Text = f[2];
    txtFal.Text =
frmLogin.convetirFechaMySQL(dr.GetString("FECHA_ALTA"));
    txtFba.Text =
frmLogin.convetirFechaMySQL(dr.GetString("FECHA_BAJA"));
    txtTP.Text = dr.GetString("TELEFONO_2");
    dr.Close();
    dr = frmLogin.con.seleccionarCatalogo("DOMICILIOS", "*", " WHERE
IDDOMICILIO=" + pBuscado.IdDomicilio.ToString());
    if (dr.Read())
    {
        txtCalle.Text = dr.GetString("CALLE");
        txtNE.Text = dr.GetString("NUMERO_EXT");

```

```

        txtNI.Text = dr.GetString("NUMERO_INT");
        txtCol.Text = dr.GetString("COLONIA");
        txtCP.Text = dr.GetString("CODIGO_POSTAL");
        txtTC.Text = dr.GetString("TELEFONO_1");
        int idTD = dr.GetInt32("IDTIPO_DOMICILIO");
        int idMunicipio = dr.GetInt32("IDMUNICIPIO_DELEGACION");
        int idEstado = dr.GetInt32("IDESTADO");
        dr.Close();
        cmbTD.Text =
frmLogin.con.buscarElemento("CAT_TIPOS_DOMICILIO", "TIPO_DOMICILIO", "
WHERE IDTIPO_DOMICILIO=" + idTD.ToString());
        cmbMun.Text =
frmLogin.con.buscarElemento("CAT_MUNICIPIOS_DELEGACIONES",
"MUNICIPIO_DELEGACION", " WHERE IDMUNICIPIO_DELEGACION=" +
idMunicipio.ToString());
        cmbEst.Text = frmLogin.con.buscarElemento("CAT_ESTADOS",
"ESTADO", " WHERE IDESTADO=" + idEstado.ToString());
    }
}
}
}
}

public void buscarDireccion()
{
    pBuscado.IdPaciente = 0;
    pBuscado.IdDomicilio = 0;
    incluirbajas = false;
    frmBuscarPaciente a = new frmBuscarPaciente();
    a.ShowDialog();
    if(pBuscado.IdPaciente > 0)
    {
        MySqlDataReader dr = frmLogin.con.seleccionarCatalogo("DOMICILIOS",
        "*", " WHERE IDDOMICILIO=" + pBuscado.IdDomicilio.ToString());
        if (dr.Read())
        {
            gpoDireccion.Enabled = false;

```

```

txtCalle.Text = dr.GetString("CALLE");
txtNE.Text = dr.GetString("NUMERO_EXT");
txtNI.Text = dr.GetString("NUMERO_INT");
txtCol.Text = dr.GetString("COLONIA");
txtCP.Text = dr.GetString("CODIGO_POSTAL");
txtTC.Text = dr.GetString("TELEFONO_1");
int idTD = dr.GetInt32("IDTIPO_DOMICILIO");
int idMunicipio = dr.GetInt32("IDMUNICIPIO_DELEGACION");
int idEstado = dr.GetInt32("IDESTADO");
dr.Close();
cmbTD.Text = frmLogin.con.buscarElemento("CAT_TIPOS_DOMICILIO",
"TIPO_DOMICILIO", " WHERE IDTIPO_DOMICILIO=" + idTD.ToString());
cmbMun.Text =
frmLogin.con.buscarElemento("CAT_MUNICIPIOS_DELEGACIONES",
"MUNICIPIO_DELEGACION", " WHERE IDMUNICIPIO_DELEGACION=" +
idMunicipio.ToString());
cmbEst.Text = frmLogin.con.buscarElemento("CAT_ESTADOS",
"ESTADO", " WHERE IDESTADO=" + idEstado.ToString());
return;
}
}
}

private void btnCan_Click(object sender, EventArgs e)
{
frmControlUsuarios.limpiarGrupo(gpoDatos);
gpoDatos.Text = "";
gpoDatos.Enabled = false;
gpoDireccion.Enabled = false;
btnAlt.Enabled = true;
btnBaj.Enabled = true;
btnCam.Enabled = true;
btnAE.Enabled = false;
btnCan.Enabled = false;
btnAE.BackgroundImage = ilAE.Images[0];
lblAE.Text = "Aceptar";
}

```

```

private void btnAE_Click(object sender, EventArgs e)
{
    if (gpoDatos.Text == "A l t a" && validarInformacion() &&
        MessageBox.Show("¿Desea dar de alta el numero de paciente " +
txtNP.Text + "?", "Alta de paciente", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
        altaPacienteDB();
    else if (gpoDatos.Text == "B a j a" && txtNP.Text.CompareTo("")!=0 &&
validarInformacion() &&
        MessageBox.Show("¿Desea dar de baja el numero de paciente " +
txtNP.Text + "?", "Baja de paciente", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
        bajaPacienteDB();
    if (gpoDatos.Text == "M o d i f i c a r" && txtNP.Text.CompareTo("")!=0 &&
validarInformacion() &&
        MessageBox.Show("¿Desea guardar los cambios realizados?", "Modificar
paciente", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.Yes)
        modificarPacienteDB();
}

```

```

public bool validarInformacion()
{
    if (txtCur.Text.Length != txtCur.MaxLength)
    {
        MessageBox.Show("El CURP debe contener 18 dígitos.", "Alta de paciente",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }

    foreach (Control c in gpoDatos.Controls)
    {
        if (c is TextBox && c.Name != txtNI.Text)
            if (c.Text.CompareTo("") == 0 && c.Name.CompareTo("txtNI") != 0)

```

```

        {
            MessageBox.Show("Solamente el campo de Número interior puede
estar vacío.", "Alta de paciente", MessageBoxButtons.OK, MessageBoxIcon.Error);
            return false;
        }
    }
    foreach (Control c in gpoDireccion.Controls)
    {
        if (c is TextBox && c.Name != txtNI.Text)
            if (c.Text.CompareTo("") == 0 && c.Name.CompareTo("txtNI") != 0)
            {
                MessageBox.Show("Solamente el campo de Número interior puede
estar vacío.", "Alta de paciente", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return false;
            }
    }
    return frmControlUsuarios.validarFechaBaja(txtFba.Text);
}

public void altaPacienteDB()
{
    if (frmLogin.con.conectar())
    {
        MySqlDataReader dr = frmLogin.con.seleccionarCatalogo("PACIENTES",
"IDPACIENTE", " WHERE NUMERO_PACIENTE=" + txtNP.Text + "");
        if (!dr.Read())//validadmos si el usuario existe
        {
            dr.Close();
            if (gpoDireccion.Enabled && !frmLogin.con.insertarPaciente(txtNP.Text,
txtNom.Text, txtApp.Text, txtApm.Text, txtCur.Text, generarFN(), txtFal.Text,
txtFba.Text, txtTP.Text, cmbTD.Text, txtCalle.Text,
txtNE.Text, txtNI.Text, txtCol.Text, cmbMun.Text, cmbEst.Text,
txtCP.Text, txtTC.Text))
                return;
        }
    }
}

```

```

        else if (!gpoDireccion.Enabled &&
!frmLogin.con.insertarPaciente(txtNP.Text, txtNom.Text, txtApp.Text, txtApm.Text,
txtCur.Text, generarFN(), pBuscado.IdDomicilio, txtFal.Text, txtFba.Text, txtTP.Text))
            return;
        frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(),
"3", "3", "A a " + txtNP.Text + " -" + frmLogin.uLogeado.Usuario,
frmLogin.fechaHoyMySQL(), frmLogin.horaMySQL());
        MessageBox.Show("El paciente " + txtNP.Text + " fue agregado
 exitosamente.", "Alta de paciente", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        btnCan_Click(null, null);
    }
    else
        MessageBox.Show("El numero de paciente " + txtNP.Text + " ya existe.",
"Alta de paciente", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
public void bajaPacienteDB()
{
    if (frmLogin.con.bajaPaciente(txtNP.Text))
    {
        frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "3",
"4", "E a " + txtNP.Text + " -" + frmLogin.uLogeado.Usuario,
frmLogin.fechaHoyMySQL(), frmLogin.horaMySQL());
        MessageBox.Show("El paciente " + txtNP.Text + " dado de baja
 exitosamente.", "Baja de paciente", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        btnCan_Click(null, null);
    }
}
public void modificarPacienteDB()
{
    if (frmLogin.con.conectar())
    {
        if (gpoDireccion.Enabled && !frmLogin.con.modificarPaciente(txtNP.Text,
txtNom.Text, txtApp.Text, txtApm.Text, txtCur.Text, generarFN(), txtFal.Text,
txtFba.Text, txtTP.Text, cmbTD.Text, txtCalle.Text,

```

```

        txtNE.Text, txtNI.Text, txtCol.Text, cmbMun.Text, cmbEst.Text,
txtCP.Text, txtTC.Text))
        return;
    else if (!gpoDireccion.Enabled &&
!frmLogin.con.modificarPaciente(txtNP.Text, txtNom.Text, txtApp.Text, txtApm.Text,
txtCur.Text, generarFN(), pBuscado.IdDomicilio, txtFal.Text, txtFba.Text, txtTP.Text))
        return;
    frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(),
"3", "5", "M a " + txtNP.Text + " -" + frmLogin.uLogeado.Usuario,
frmLogin.fechaHoyMySQL(), frmLogin.horaMySQL());
    MessageBox.Show("El paciente " + txtNP.Text + " fue modificado
 exitosadamente.", "Modificar paciente", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    btnCan_Click(null, null);
}
}

public string generarFN()
{
    cmbMes.SelectedIndex= cmbMes.Items.IndexOf(cmbMes.Text);
    return cmbAnho.Text + "-" + (cmbMes.SelectedIndex + 1) + "-" + cmbDia.Text;
}

private void btnBaj_Click(object sender, EventArgs e)
{
    gpoDatos.Text = "B a j a";
    gpoDatos.Enabled = false;
    gpoDireccion.Enabled = false;
    btnAlt.Enabled = false;
    btnBaj.Enabled = true;
    btnCam.Enabled = false;
    btnAE.Enabled = true;
    btnCan.Enabled = true;
    btnAE.BackgroundImage = ilAE.Images[1];
    lblAE.Text = "Eliminar";
    Refresh();
    buscarPaciente();
}

```

```

}

private void btnCam_Click(object sender, EventArgs e)
{
    gpoDatos.Text = "M o d i f i c a r";
    gpoDatos.Enabled = true;
    gpoDireccion.Enabled = true;
    btnAlt.Enabled = false;
    btnBaj.Enabled = false;
    btnCam.Enabled = true;
    btnAE.Enabled = true;
    btnCan.Enabled = true;
    btnAE.BackgroundImage = ilAE.Images[0];
    lblAE.Text = "Aceptar";
    Refresh();
    buscarPaciente();
}

private void frmControlPacientes_Load(object sender, EventArgs e)
{
    frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "3",
"1", "I a CP -" + frmLogin.uLogeado.Usuario, frmLogin.fechaHoyMySQL(),
frmLogin.horaMySQL());
}

private void frmControlPacientes_FormClosing(object sender,
FormClosingEventArgs e)
{
    frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "3",
"2", "S de CP -" + frmLogin.uLogeado.Usuario, frmLogin.fechaHoyMySQL(),
frmLogin.horaMySQL());
}
}
}

```



“frmControlUsuarios.cs”

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Text.RegularExpressions;//REGEX
using MySql.Data.MySqlClient;

namespace Health_Card_Manager
{
    public partial class frmControlUsuarios : Form
    {
        public static cUsuario uBuscado = new cUsuario();
        public static bool incluirBajas;
        public frmControlUsuarios()
        {
            InitializeComponent();
            inicializarListas();
        }

        private void btnAlt_Click(object sender, EventArgs e)
        {
            modificarControles(true, "A l t a", true, true, "Aceptar", false, false, false, 0);
            elegirIndexGrupo(gpoDatos);
        }

        public void elegirIndexGrupo(GroupBox grupo)
        {
            cmbEsp.SelectedIndex = 0;
            cmbEsp.Text = cmbEsp.SelectedItem.ToString();
            cmbSub.SelectedIndex = 0;
            cmbSub.Text = cmbSub.SelectedItem.ToString();
            cmbClu.SelectedIndex = 0;
        }
    }
}
```

```

        cmbClu.Text = cmbClu.SelectedItem.ToString();
        cmbRol.SelectedIndex = 3;
        cmbRol.Text = cmbRol.SelectedItem.ToString();
        habilitarCajas(false, false, false, true, true);
    }

    public void inicializarListas()
    {
        if(frmLogin.con.conectar())
        {
            frmLogin.con.asignarCatalogoLista("CAT_ESPECIALIDADES",
"ESPECIALIDAD", cmbEsp, false);
            frmLogin.con.asignarCatalogoLista("CAT_ESPECIALIDADES",
"ESPECIALIDAD", cmbSub, false);
            frmLogin.con.asignarCatalogoLista("CAT_CLUES", "CLUES", cmbClu,
false);
            frmLogin.con.asignarCatalogoLista("CAT_ROLES", "ROL", cmbRol, false);
        }
    }

    private void btnCan_Click(object sender, EventArgs e)
    {
        modificarControles(false, "", false, false, "Aceptar", true, true, true, 0);
        limpiarGrupo(gpoDatos);
        uBuscado.Usuario = "";
        //bug en cancelar
        cmbRol.Text = "";
        habilitarCajas(false, false, false, false, false);
        gpoDatos.Size = new System.Drawing.Size(299, 355);
    }

    public static void limpiarGrupo(GroupBox grupo)
    {
        foreach (Control c in grupo.Controls)
        {

```

```

if (c is CheckBox)
    ((CheckBox)c).Checked = false;
else if (c is TextBox || c is ComboBox)
    c.Text = "";
else if (c is GroupBox)
    limpiarGrupo(((GroupBox)c));
}
}

```

```

private void cmbRol_SelectedIndexChanged(object sender, EventArgs e)
{
    int caseSwitch = cmbRol.SelectedIndex;
    gpoModulos.Enabled = false;
    switch (caseSwitch)
    {
        case 0://MASTER
            habilitarCajas(true,true,true,true,true);
            break;
        case 1://ADMINISTRADOR
            habilitarCajas(false,true,true,true,true);
            break;
        case 2://MEDICO
            if (txtCed.Text != "" && txtCed.Text.Length==7)
                habilitarCajas(false, true, true, false, false);
            else
            {
                MessageBox.Show("Para elegir éste rol, ingresar el campo Cédula (7
dígitos).", "Elección de rol", MessageBoxButtons.OK, MessageBoxIcon.Information);
                cmbRol.SelectedIndex = 3;
            }
            break;
        case 3://OPERADOR
            habilitarCajas(false,false,false,true,true);
            break;
        case 4://PERSONALIZADO
            habilitarCajas(false, false, false, false, false);
            gpoModulos.Enabled = true;

```

```

        break;
    }
}

public void habilitarCajas(bool cu, bool cp, bool ds, bool c, bool r)
{
    chkCU.Checked = cu;
    chkCP.Checked = cp;
    chkDS.Checked = ds;
    chkC.Checked = c;
    chkR.Checked = r;
}

private void cmbRol_TextUpdate(object sender, EventArgs e)
{valorOriginal(cmbRol);}

private void cmbClu_TextUpdate(object sender, EventArgs e)
{valorOriginal(cmbClu);}

private void cmbSub_TextUpdate(object sender, EventArgs e)
{valorOriginal(cmbSub);}

private void cmbEsp_TextUpdate(object sender, EventArgs e)
{valorOriginal(cmbEsp);}

public void valorOriginal(ComboBox c)
{
    if (gpoDatos.Text.CompareTo("B a j a")>0)
    {
        if (c.Name.ToString().CompareTo("cmbRol") == 0)
            c.SelectedIndex = 3;
        else
            c.SelectedIndex = 0;
    }
}
}

```

```

private void txtCur_KeyPress(object sender, KeyPressEventArgs e)
{evitarTeclas(e,false, false, true);}

public static void evitarTeclas(KeyPressEventArgs e, bool numeros, bool letras,
bool espaciosBlancos)
{
    if (Char.IsNumber(e.KeyChar))
        e.Handled = numeros;
    else if (Char.IsLetter(e.KeyChar))
        e.Handled = letras;
    else if (Char.IsWhiteSpace(e.KeyChar))
        e.Handled = espaciosBlancos;
    else if (Char.IsControl(e.KeyChar))
        e.Handled = false;
    else if (Char.IsSeparator(e.KeyChar))
        e.Handled = false;
    else
        e.Handled = true;
}

private void txtCed_KeyPress(object sender, KeyPressEventArgs e)
{evitarTeclas(e, false, true, true);}

private void txtNom_KeyPress(object sender, KeyPressEventArgs e)
{evitarTeclas(e, true, false, false);}

private void button1_Click(object sender, EventArgs e)
{generarUsuario(txtNom.Text, txtCur.Text, txtUsu, txtCon);}

public void generarUsuario(string n, string c, TextBox usuario, TextBox
contrasena)
{
    if (n == "")
        MessageBox.Show("Es necesario ingresar el campo Nombre(s).", "Generar
usuario", MessageBoxButtons.OK, MessageBoxIcon.Error);
    else if (c == "" || c.Length < 18)

```

```

        MessageBox.Show("Es necesario ingresar el campo CURP (18 dígitos).",
"Generár usuario", MessageBoxButtons.OK, MessageBoxIcon.Error);
    else
    {
        string cadusu = n.Split(' ')[0] + c.Substring(6,4);
        string cadcon = cadusu;
        usuario.Text = sinAcentos(cadusu);
        contrasena.Text = sinAcentos(cadcon);
    }
}

public static string sinAcentos(string palabra)
{
    Regex replace_a_Accents = new Regex("[Á]", RegexOptions.Compiled);
    Regex replace_e_Accents = new Regex("[É]", RegexOptions.Compiled);
    Regex replace_i_Accents = new Regex("[Í]", RegexOptions.Compiled);
    Regex replace_o_Accents = new Regex("[Ó]", RegexOptions.Compiled);
    Regex replace_u_Accents = new Regex("[Ú]", RegexOptions.Compiled);
    palabra = replace_a_Accents.Replace(palabra, "a");
    palabra = replace_e_Accents.Replace(palabra, "e");
    palabra = replace_i_Accents.Replace(palabra, "i");
    palabra = replace_o_Accents.Replace(palabra, "o");
    palabra = replace_u_Accents.Replace(palabra, "u");
    return palabra;
}

public bool validarInformacion()
{
    if (txtCur.Text.Length != txtCur.MaxLength)
    {
        MessageBox.Show("El CURP debe contener 18 dígitos.", "Alta de usuario",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
    else if (chkCU.Checked == false && chkCP.Checked == false &&
chkDS.Checked == false && chkC.Checked == false && chkR.Checked == false)
    {

```

```

        MessageBox.Show("Debe estar seleccionado por lo menos un módulo.",
"Alta de usuario", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }

    foreach (Control c in gpoDatos.Controls)
        if (c is TextBox && c.Name != txtCed.Name)
            if (c.Text.CompareTo("") == 0 && c.Name.CompareTo("txtFba")!=0 &&
c.Name.CompareTo("txtFal")!=0)
                {
                    MessageBox.Show("Solamente el campo de Cédula puede estar
vacío.", "Alta de usuario", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    return false;
                }
            if (gpoDatos.Text.CompareTo("M o d i f i c a r") == 0)
                return validarFechaBaja(txtFba.Text);

    return true;
}

public static bool validarFechaBaja(string fecha)
{
    try
    {
        string[] f = fecha.Split('-');
        if (fecha.CompareTo("") > 0 && f.Length == 3 && int.Parse(f[0]) > 1000 &&
int.Parse(f[1]) < 13 && int.Parse(f[2]) < 32)
            return true;
        MessageBox.Show("Valor invalido (yyyy-mm-dd).", "Fecha de baja",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + " (yyyy-mm-dd)", "Fecha de baja",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```

        return false;
    }

}

private void btnAE_Click(object sender, EventArgs e)
{
    if (gpoDatos.Text == "A l t a" && validarInformacion() &&
        MessageBox.Show("¿Desea dar de alta al usuario " + txtUsu.Text + "?",
            "Alta de usuario", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
        DialogResult.Yes)
        altaUsuarioDB();
    else if (gpoDatos.Text == "B a j a" &&
        MessageBox.Show("¿Desea dar de baja al usuario " + txtUsu.Text + "?",
            "Baja de usuario", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
        DialogResult.Yes)
        bajaUsuarioDB();
    else if (gpoDatos.Text == "M o d i f i c a r"
        if (validarInformacion() && MessageBox.Show("¿Desea guardar los cambios
            realizados?", "Modificar Usuario", MessageBoxButtons.YesNo,
            MessageBoxIcon.Question) == DialogResult.Yes)
            modificarUsuarioDB();
        else
            return;
        gpoDatos.Size = new System.Drawing.Size(299, 355);
    }

public void bajaUsuarioDB()
{
    if (frmLogin.con.bajaUsuario(uBuscado.IdUsuario))
    {
        frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "2",
            "4", "E a " + txtUsu.Text + " -" + frmLogin.uLogeado.Usuario,
            frmLogin.fechaHoyMySQL(), frmLogin.horaMySQL());
    }
}

```



```

        MessageBox.Show("El usuario " + txtUsu.Text + " dado de baja
 exitosamente.", "Baja de usuario", MessageBoxButtons.OK,
 MessageBoxIcon.Information);
        uBuscado.Usuario = "";
        btnCan_Click(null,null);
    }
}

public void altaUsuarioDB()
{
    if (frmLogin.con.conectar())
    {
        MySqlDataReader dr = frmLogin.con.seleccionarCatalogo("USUARIOS",
 "IDUSUARIO", " WHERE USUARIO=" + txtUsu.Text + "");
        if (dr.Read())//validamos si el usuario existe
            MessageBox.Show("El usuario " + txtUsu.Text + " ya existe.", "Alta de
 usuario", MessageBoxButtons.OK, MessageBoxIcon.Error);
        else if (frmLogin.con.insertarUsuario(txtCed.Text, txtCur.Text, txtNom.Text,
 txtApp.Text, txtApm.Text, cmbEsp.SelectedIndex + 1,
        cmbSub.SelectedIndex + 1, cmbClu.SelectedIndex + 1,
        cmbRol.SelectedIndex + 1, txtUsu.Text, txtCon.Text, frmLogin.fechaHoyMySQL(),
 "9999-12-31", true))
        {
            dr = frmLogin.con.seleccionarCatalogo("USUARIOS", "IDUSUARIO", "
 WHERE USUARIO=" + txtUsu.Text + "");//buscamos el id que se asigno al usuario
            if (dr.Read())
            { //insertamos en la DB los modulos a los que tiene acceso
                if (frmLogin.con.insertarUsuariosModulos(dr.GetInt32(0),
                chkCU.Checked, chkCP.Checked, chkDS.Checked, chkC.Checked, chkR.Checked))
                {
                    frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "2", "3", "A a "
                    + txtUsu.Text + " -" + frmLogin.uLogeado.Usuario, frmLogin.fechaHoyMySQL(),
                    frmLogin.horaMySQL());
                }
            }
        }
    }
}

```



```

a.ShowDialog();
if (uBuscado.Usuario.CompareTo("")>0)
    if (frmLogin.con.conectar())
    {
        MySqlDataReader dr = frmLogin.con.seleccionarCatalogo("USUARIOS",
        "*", " WHERE USUARIO=" + uBuscado.Usuario + "");
        if (dr.Read())
        {
            uBuscado.BDusuarioTOcUsuario(dr);
            dr.Close();
            modificarControles(false, "B a j a", true, true, "Eliminar", false, false,
false, 1);
            cUsuarioTOcontroles(uBuscado);
            dr = frmLogin.con.seleccionarCatalogo("USUARIOS_MODULOS",
"IDMODULO", " WHERE IDUSUARIO=" + uBuscado.IdUsuario + "");
            cargarModulos(dr);
        }
        else
            MessageBox.Show("No existe el usuario " + uBuscado.Usuario + ". ",
"Iniciar Sesión", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

public void cargarModulos(MySqlDataReader dr)
{
    habilitarCajas(false,false,false,false,false);
    while (dr.Read())
    {
        int caseSwitch = dr.GetInt32(0);
        gpoModulos.Enabled = false;
        switch (caseSwitch)
        {
            case 2:
                chkCU.Checked=true;
                break;
            case 3:

```

```

        chkCP.Checked=true;
        break;
    case 4:
        chkDS.Checked=true;
        break;
    case 5:
        chkC.Checked=true;
        break;
    case 6:
        chkR.Checked=true;
        break;
    }
}
}

```

```

private void btnCam_Click(object sender, EventArgs e)
{
    incluirBajas = true;
    frmBuscarUsuario a = new frmBuscarUsuario();
    a.ShowDialog();
    if (uBuscado.Usuario.CompareTo("") > 0)
        if (frmLogin.con.conectar())
        {
            MySqlDataReader dr = frmLogin.con.seleccionarCatalogo("USUARIOS",
            "*", " WHERE USUARIO=" + uBuscado.Usuario + "");
            if (dr.Read())
            {
                uBuscado.BDusuarioTOcUsuario(dr);
                dr.Close();
                modificarControles(true, "M o d i f i c a r", true, true, "Aceptar", false,
                false, false, 0);
                cUsuarioTOcontroles(uBuscado);
                dr = frmLogin.con.seleccionarCatalogo("USUARIOS_MODULOS",
                "IDMODULO", " WHERE IDUSUARIO=" + uBuscado.IdUsuario + "");
                cargarModulos(dr);
                gpoDatos.Size = new System.Drawing.Size(299,412);
            }
        }
}

```

```

    }
    else
        MessageBox.Show("No existe el usuario " + uBuscado.Usuario + ". ",
            "Iniciar sesión", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```

public void cUsuarioTOcontroles(cUsuario usuario)
{
    this.Refresh();
    txtCed.Text = usuario.Cedula;
    txtCur.Text = usuario.Curp;
    txtNom.Text = usuario.Nombre;
    txtApp.Text = usuario.ApPaterno;
    txtApm.Text = usuario.ApMaterno;
    cmbEsp.SelectedIndex = usuario.IdEspecialidad - 1;
    cmbEsp.Text = cmbEsp.SelectedItem.ToString();
    cmbSub.SelectedIndex = usuario.IdsubEspecialidad - 1;
    cmbSub.Text = cmbSub.SelectedItem.ToString();
    this.Refresh();
    cmbClu.SelectedIndex = usuario.IdClues - 1;
    cmbClu.Text = cmbClu.SelectedItem.ToString();
    cmbRol.SelectedIndex = usuario.IdRol - 1;
    cmbRol.Text = cmbRol.SelectedItem.ToString();
    txtUsu.Text = usuario.Usuario;
    txtCon.Text = usuario.Contrasena;
    txtFal.Text = frmLogin.conveterFechaMySQL(usuario.FechaAlta);
    txtFba.Text = frmLogin.conveterFechaMySQL(usuario.FechaBaja);
    chkAct.Checked = usuario.Activado;
    this.Refresh();
}

```

```

public void modificarUsuarioDB()
{
    if (frmLogin.con.conectar())
        if (frmLogin.con.modificarUsuario(uBuscado.IdUsuario, txtCed.Text,
            txtCur.Text, txtNom.Text, txtApp.Text, txtApm.Text, cmbEsp.SelectedIndex + 1,

```

```

        cmbSub.SelectedIndex + 1, cmbClu.SelectedIndex + 1,
        cmbRol.SelectedIndex + 1, txtUsu.Text, txtCon.Text, txtFal.Text, txtFba.Text,
        chkAct.Checked))
            if (frmLogin.con.eliminarUsuariosModulos(uBuscado.IdUsuario))
            {
                this.Refresh();
                if (frmLogin.con.insertarUsuariosModulos(uBuscado.IdUsuario,
        chkCU.Checked, chkCP.Checked, chkDS.Checked, chkC.Checked, chkR.Checked))
                {
                    frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "2", "5", "M a "
        + txtUsu.Text + " -" + frmLogin.uLogeado.Usuario, frmLogin.fechaHoyMySQL(),
        frmLogin.horaMySQL());
                    MessageBox.Show("El usuario " + txtUsu.Text + " fue actualizado
        exitosamente.", "Alta de Usuario", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
                    btnCan_Click(null, null);
                }
            }
    }
}

```

```

    public void modificarControles(bool hGpoDat, string tGpoDat, bool hBtnCan, bool
        hBtnAE, string tLbIAE,
            bool hBtnAlt, bool hBtnBaj, bool hBtnCam, int indexImage)
    {
        gpoDatos.Enabled = hGpoDat;
        gpoDatos.Text = tGpoDat;
        btnCan.Enabled = hBtnCan;
        btnAE.Enabled = hBtnAE;
        lbIAE.Text = tLbIAE;
        btnAlt.Enabled = hBtnAlt;
        btnBaj.Enabled = hBtnBaj;
        btnCam.Enabled = hBtnCam;
        btnAE.BackgroundImage = ilAE.Images[indexImage];
        Refresh();
    }
}

```

```

    }

    private void frmControlUsuarios_Load(object sender, EventArgs e)
    {
        frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "2",
        "1", "I a CU -" + frmLogin.uLogeado.Usuario, frmLogin.fechaHoyMySQL(),
        frmLogin.horaMySQL());
    }

    private void frmControlUsuarios_FormClosing(object sender,
    FormClosingEventArgs e)
    {
        frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "2",
        "2", "S de CU -" + frmLogin.uLogeado.Usuario, frmLogin.fechaHoyMySQL(),
        frmLogin.horaMySQL());
    }
}
}
}

```

“frmCredencializacion.cs”

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using System.Collections;

namespace Health_Card_Manager
{
    public partial class frmCredencializacion : Form
    {
        public static int NP = 0;
        public static int idP = 0;
        Queue<string> elementos = new Queue<string>();

        public frmCredencializacion()
        {
            InitializeComponent();
        }

        private void frmCredencializacion_Load(object sender, EventArgs e)
        {
            frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "5",
            "1", "I a C -" + frmLogin.uLogeado.Usuario, frmLogin.fechaHoyMySQL(),
            frmLogin.horaMySQL());
            buscarPacienteS();
        }

        private void btnAce_Click(object sender, EventArgs e)
        {
            asignarCredencial();
        }
    }
}
```



```

public void asignarCredencial()
{
    try
    {
        string SN = "";
        int res = ACR120U.ACR120_Open(0);
        if (res == 0)
        {
            long sto = 0;
            byte[] ResultSN = new byte[11];
            byte LengthTag = 0x00; ;
            byte[] TagType = new byte[51];

            res = ACR120U.ACR120_Select(0, ref TagType[0], ref LengthTag, ref
ResultSN[0]);
            if (res == 62536)
            {
                MessageBox.Show("Colocar credencial sobre el lector ACR120U",
"Seleccionar credencial", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }
            else if (res < 0)
            {
                MessageBox.Show(ACR120U.GetErrMsg(res), "Seleccionar
credencial", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }
            else
            {

                string tipoCredencial = ACR120U.GetTagType1(TagType[0]);
                if (tipoCredencial.CompareTo("Mifare 4K") == 0 ||
tipoCredencial.CompareTo("Mifare 1K") == 0)
                {

```

```

for (int ctr = 0; ctr < LengthTag; ctr++)
    SN = SN + string.Format("{0:X2} ", ResultSN[ctr]);
string noexiste=frmLogin.con.buscarElemento("CREDENCIALES",
"IDCREDENCIAL", " WHERE NUMERO_CREDENCIAL=" + SN + " AND
IDPACIENTE<>" + idP.ToString() + " AND FECHA_BAJA>" +
frmLogin.fechaHoyMySQL() + "" );
if (noexiste.CompareTo("") == 0)
{
    byte[] key ={ 255, 255, 255, 255, 255, 255 };
    int sector;
    for (sector = 0; sector < 39; sector++)
    {
        res = ACR120U.ACR120_Login(0, Convert.ToByte(sector),
Convert.ToInt16(ACR120U.ACR120_LOGIN_KEYTYPE_A), Convert.ToByte(sto), ref
key[0]); //G_retcod = ACR120U.ACR120_Login(G_rHandle, 1,
ACR120U.KEYTYPES.ACR120_LOGIN_KEYTYPE_A, Nothing, G_pKey(0))
        if (res != 0)
            break;
    }

    if (res < 0)
        MessageBox.Show(ACR120U.GetErrMsg(res), "Verificando
clave del sector " + sector.ToString());
    else if (res > 0)
    {
        if (sector == 0)
            MessageBox.Show(res.ToString() + ": La clave del sector
no corresponde", "Verificando clave del sector " + sector.ToString());
        else
            MessageBox.Show(res.ToString() + ": La credencial " + SN
+ " fue retirada del \n lector o la clave del sector no corresponde", "Verificando clave
del sector " + sector.ToString());
    }
    else
    {

```

```

int modulo = 0;
int resW = 0;
byte[] ele = new byte[16];

for (sector = 1; sector < 39; sector++)
{
    res = ACR120U.ACR120_Login(0, Convert.ToByte(sector),
Convert.ToInt16(ACR120U.ACR120_LOGIN_KEYTYPE_A), Convert.ToByte(sto), ref
key[0]); //G_retcod = ACR120U.ACR120_Login(G_rHandle, 1,
ACR120U.KEYTYPES.ACR120_LOGIN_KEYTYPE_A, Nothing, G_pKey(0))
    if (res != 0)
    {
        if (res < 0)
            MessageBox.Show(ACR120U.GetErrMsg(res),
"Validando clave del sector " + sector.ToString());
        else if (res > 0)
        {
            if (sector == 1)
                MessageBox.Show(res.ToString() + ": La clave del
sector no corresponde", "Validando clave del sector " + sector.ToString());
            else
                MessageBox.Show(res.ToString() + ": La credencial
" + SN + " fue retirada del \n lector o la clave del lector no corresponde", "Verificando
clave del sector " + sector.ToString());
        }
        break;
    }
    if (sector < 32)
    {
        string elestr = "";
        for (modulo = 0; modulo < 3; modulo++)
            if (elementos.Count > 0)
            {
                elestr = elementos.Dequeue();
                ele = S2B(elestr);
                resW = ACR120U.ACR120_Write(0,
Convert.ToByte(sector * 4 + modulo), ref ele[0]);

```

```

        if (resW != 0)
            goto Error;
    }
}
else
{
    for (modulo = 0; modulo < 15; modulo++)
        if (elementos.Count > 0)
        {
            ele = S2B(elementos.Dequeue());
            resW = ACR120U.ACR120_Write(0,
Convert.ToByte(sector * 4 + modulo), ref ele[0]);
            if (resW != 0)
                goto Error;
        }
    }
}
Error:
    if (res == 0 && resW < 0)
        MessageBox.Show(ACR120U.GetErrMsg(res),
"Escribiendo en el sector " + sector.ToString());
    else if (res == 0 && resW > 0)
    {
        MessageBox.Show(res.ToString() + ": Error inesperado al
escribir", "Escribiendo en el sector " + sector.ToString());
    }
}
}
else
{
    MessageBox.Show("La credencial " + SN + " ya fue asignada a
otro paciente", "Validando
credencial", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    return;
}
}
}

```

```

        else
            MessageBox.Show("Tarjetas " + tipoCredencial + " no son
soportadas por la aplicacion", "Tarjeta Mifare Classic 4K", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
    }
    ACR120U.ACR120_Close(0);
    frmLogin.con.insertarCredencial(idP.ToString(), SN);
    lbINC.Text = SN;
    frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(),
"5", "3", "A " + SN + " -" + frmLogin.uLogeado.Usuario, frmLogin.fechaHoyMySQL(),
frmLogin.horaMySQL());
    MessageBox.Show("La credencial " + SN + " fue asignada
exitosamente", "Credencializacion", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    borrarInfo();
}
else
    MessageBox.Show(ACR120U.GetErrMsg(res), "Conectar el lector
ACR120U", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Error de credencializacion",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

private void button1_Click(object sender, EventArgs e)
{
    buscarPacienteS();
}

public void borrarInfo()
{
    lstAle2.Items.Clear();
}

```

```

lstDis2.Items.Clear();
lstEnf2.Items.Clear();

btnAce.Visible=false;
lblNom.Text = "";
lblINP.Text = "";
NP = 0;
idP = 0;
lblINC.Text = "";
lblSex.Text = "";
lblSan.Text = "";
lblViv.Text = "";
lblTel.Text = "";
lblNiv.Text = "";
lblRel.Text = "";
lblGru.Text = "";

}

public void buscarPacienteS()
{
    borrarInfo();
    frmBuscarPacienteS a = new frmBuscarPacienteS(false);
    a.ShowDialog();
    if (NP > 0 && idP > 0)
    {
        MySqlDataReader dr = frmLogin.con.seleccionarCatalogo("PACIENTES",
        "*", " WHERE IDPACIENTE=" + idP.ToString());
        if (dr.Read())
        {
            lblINP.Text = dr.GetString("NUMERO_PACIENTE");
            lblNom.Text = dr.GetString("NOMBRE") + " " +
            dr.GetString("APELLIDO_PATERNO") + " " + dr.GetString("APELLIDO_MATERNO");
            if (!dr.IsDBNull(8))
            {
                string idNiv = dr.GetString("IDNIVEL_SOCIOECONOMICO");

```

```

        string idViv = dr.GetString("IDVIVIENDA");
        string idGru = dr.GetString("IDGRUPO_ETNICO");
        string idRel = dr.GetString("IDRELIGION");
        string idSex = dr.GetString("IDSEXO");
        string idSan = dr.GetString("IDSANGRE");
        string idDom = dr.GetString("IDDOMICILIO");
        dr.Close();
        lblNiv.Text =
frmLogin.con.buscarElemento("CAT_NIVELES_SOCIOECONOMICOS",
"NIVEL_SOCIOECONOMICO", " WHERE IDNIVEL_SOCIOECONOMICO=" + idNiv);
        lblViv.Text = frmLogin.con.buscarElemento("CAT_VIVIENDAS",
"VIVIENDA", " WHERE IDVIVIENDA=" + idViv);
        lblGru.Text =
frmLogin.con.buscarElemento("CAT_GRUPOS_ETNICOS", "GRUPO_ETNICO", "
WHERE IDGRUPO_ETNICO=" + idGru);
        lblRel.Text = frmLogin.con.buscarElemento("CAT_RELIGIONES",
"RELIGION", " WHERE IDRELIGION=" + idRel);
        lblSex.Text = frmLogin.con.buscarElemento("CAT_SEXOS", "SEXO", "
WHERE IDSEXO=" + idSex);
        lblSan.Text = frmLogin.con.buscarElemento("CAT_SANGRE",
"SANGRE", " WHERE IDSANGRE=" + idSan);
        lblTel.Text = frmLogin.con.buscarElemento("DOMICILIOS",
"TELEFONO_1", " WHERE IDDOMICILIO=" + idDom);

        frmLogin.con.leerPacienteEDA(idP.ToString(), lstEnf2, lstDis2, lstAle2);

        lblINC.Text = frmLogin.con.buscarElemento("CREDENCIALES",
"NUMERO_CREDENCIAL", " WHERE IDPACIENTE=" + idP + " AND FECHA_BAJA>"
+ frmLogin.fechaHoyMySQL() + "");
        btnAce.Visible=true;
        generarCola();

    }
    else
    {
        if (!dr.IsClosed)
            dr.Close();
    }
}

```

```

        borrarInfo();
    }

    if (!dr.IsClosed)
        dr.Close();
    }
}

private void cmbSex_SelectedIndexChanged(object sender, EventArgs e)
{
}

public void generarCola()
{
    elementos = new Queue<string>();
    string c="";
    c += lbINP.Text;
    c += "$";
    c += lbINom.Text;
    c += "$";
    c += lbISex.Text;
    c += "$";
    c += lbISan.Text;
    c += "$";
    c += lbINiv.Text;
    c += "$";
    c += lbIViv.Text;
    c += "$";
    c += lbIGru.Text;
    c += "$";
    c += lbIRel.Text;
    c += "$";
    for (int i = 0; i < lstEnf2.Items.Count; i++)

```



```

{
    c += lstEnf2.Items[i].ToString();
    if ((i+1) < lstEnf2.Items.Count)
        c += ",";
}
c += "$";
for (int i = 0; i < lstDis2.Items.Count; i++)
{
    c += lstDis2.Items[i].ToString();
    if ((i + 1) < lstDis2.Items.Count)
        c += ",";
}
c += "$";
for (int i = 0; i < lstAle2.Items.Count; i++)
{
    c += lstAle2.Items[i].ToString();
    if ((i + 1) < lstAle2.Items.Count)
        c += ",";
}
c += "$";
c += lbITel.Text;
c += "$";
int cads = c.Length / 16;
int m= c.Length % 16;
int j;
for (j = 0; j < (cads * 16); j += 16)
{
    elementos.Enqueue(c.Substring(j, 16));
}
if (m > 0)
{
    elementos.Enqueue(c.Substring(j, c.Length - j));
}
elementos.Enqueue("#####");
}

```

```
public byte[] S2B(string s)
```

```
{
    System.Text.UTF8Encoding code = new System.Text.UTF8Encoding();
    return code.GetBytes(s);
}

private void frmCredencializacion_FormClosing(object sender,
FormClosingEventArgs e)
{
    frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "5",
"2", "S de C -" + frmLogin.uLogeado.Usuario, frmLogin.fechaHoyMySQL(),
frmLogin.horaMySQL());
}
}
}
```

"frmDatosSanidad.cs"

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Health_Card_Manager
{
    public partial class frmDatosSanidad : Form
    {
        public static int NP=0;
        public static int idP = 0;

        public frmDatosSanidad()
        {
            InitializeComponent();
            inicializarControles();
        }

        public void inicializarControles()
        {
            if (frmLogin.con.conectar())
            {
                frmLogin.con.asignarCatalogoLista("CAT_NIVELES_SOCIOECONOMICOS",
                "NIVEL_SOCIOECONOMICO", cmbNiv, false);
                frmLogin.con.asignarCatalogoLista("CAT_VIVIENDAS", "VIVIENDA",
                cmbViv, false);
                frmLogin.con.asignarCatalogoLista("CAT_GRUPOS_ETNICOS",
                "GRUPO_ETNICO", cmbGru, false);
                frmLogin.con.asignarCatalogoLista("CAT_RELIGIONES", "RELIGION",
                cmbRel, false);
            }
        }
    }
}
```

```

frmLogin.con.asignarCatalogoLista("CAT_SEXOS", "SEXO", cmbSex, false);
frmLogin.con.asignarCatalogoLista("CAT_SANGRE", "SANGRE", cmbSan,
false);
    frmLogin.con.asignarCatalogoLista("CAT_GRUPOS_ENFERMEDADES",
"GRUPO_ENFERMEDAD", cmbGE, false);
    frmLogin.con.asignarCatalogoLista("CAT_GRUPOS_DISCAPACIDADES",
"GRUPO_DISCAPACIDAD", cmbGD, false);

    MySqlDataReader dr =
frmLogin.con.seleccionarCatalogo("CAT_ALERGIAS", "ALERGIA", " ");
    while (dr.Read())
        lstAle1.Items.Add(dr.GetString("ALERGIA"));
    dr.Close();
    if(lstAle1.Items.Count>0)
        lstAle1.SelectedIndex = 0;
    sel(gpoDatos);
}
}

public void sel(GroupBox grupo)
{
    foreach (Control c in grupo.Controls)
    {
        if (c is ComboBox)
        {
            if (((ComboBox)c).Items.Count > 0)
            {
                ((ComboBox)c).SelectedIndex = 0;
                c.Text = ((ComboBox)c).SelectedItem.ToString();
            }
        }
        else if (c is GroupBox)
            sel((GroupBox)c);
    }
}

private void btnAce_Click(object sender, EventArgs e)

```

```

    {
        if (NP > 0 && MessageBox.Show("¿Desea guardar los cambios realizados?",
"Modificar datos de sanidad", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
== DialogResult.Yes)
        {
            modificarPacienteBD();
        }
    }
    public void borrarInfo()
    {
        sel(gpoDatos);
        lstAle2.Items.Clear();
        lstDis2.Items.Clear();
        lstEnf2.Items.Clear();
    }

    public void modificarPacienteBD()
    {
        if (frmLogin.con.conectar())
        {
            if (!frmLogin.con.eliminarPacienteEDA(idP.ToString()) ||
!frmLogin.con.modificarPacienteDS(idP.ToString(), cmbNiv.Text, cmbViv.Text,
cmbGru.Text, cmbRel.Text, cmbSex.Text, cmbSan.Text)
                ||
!frmLogin.con.ingresarPacienteEDA(idP.ToString(),lstEnf2,lstDis2,lstAle2))
                return;

            frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "4",
"5", "M a " + txtNP.Text + " -" + frmLogin.uLogeado.Usuario,
frmLogin.fechaHoyMySQL(), frmLogin.horaMySQL());
            MessageBox.Show("El paciente " + txtNP.Text + " fue modificado
exitosamente.", "Modificar paciente", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            borrarInfo();
            borrarNomNP();
        }
    }

```

```

}

private void cmbGE_SelectedIndexChanged(object sender, EventArgs e)
{
    lstEnf1.Items.Clear();
    int idGE =
frmLogin.con.buscarIdElemento("CAT_GRUPOS_ENFERMEDADES","IDGRUPO_EN
FERMEDAD"," WHERE GRUPO_ENFERMEDAD=" + cmbGE.Text + "");
    MySqlDataReader dr =
frmLogin.con.seleccionarCatalogo("CAT_ENFERMEDADES","ENFERMEDAD","
WHERE IDGRUPO_ENFERMEDAD=" + idGE .ToString() + "");
    while (dr.Read())
        lstEnf1.Items.Add(dr.GetString("ENFERMEDAD"));
    dr.Close();
    if(lstEnf1.Items.Count>0)
        lstEnf1.SelectedIndex = 0;
}

private void cmbGD_SelectedIndexChanged(object sender, EventArgs e)
{
    lstDis1.Items.Clear();
    int idGD =
frmLogin.con.buscarIdElemento("CAT_GRUPOS_DISCAPACIDADES",
"IDGRUPO_DISCAPACIDAD", " WHERE GRUPO_DISCAPACIDAD=" + cmbGD.Text
+ "");
    MySqlDataReader dr =
frmLogin.con.seleccionarCatalogo("CAT_DISCAPACIDADES", "DISCAPACIDAD", "
WHERE IDGRUPO_DISCAPACIDAD=" + idGD.ToString() + "");
    while (dr.Read())
        lstDis1.Items.Add(dr.GetString("DISCAPACIDAD"));
    dr.Close();
    if(lstDis1.Items.Count>0)
        lstDis1.SelectedIndex = 0;
}

private void cmbNiv_TextUpdate(object sender, EventArgs e)
{ if(cmbNiv.Items.Count>0)

```

```

        cmbNiv.SelectedIndex = 0; }

private void cmbSex_TextUpdate(object sender, EventArgs e)
{ if(cmbSex.Items.Count>0)
    cmbSex.SelectedIndex = 0; }

private void cmbViv_TextUpdate(object sender, EventArgs e)
{ if(cmbViv.Items.Count>0)
    cmbViv.SelectedIndex = 0; }

private void cmbSan_TextUpdate(object sender, EventArgs e)
{ if(cmbSan.Items.Count>0)
    cmbSan.SelectedIndex = 0; }

private void cmbRel_TextUpdate(object sender, EventArgs e)
{ if(cmbRel.Items.Count>0)
    cmbRel.SelectedIndex = 0; }

private void cmbGE_TextUpdate(object sender, EventArgs e)
{ if(cmbGE.Items.Count>0)
    cmbGE.SelectedIndex = 0; }

private void cmbGD_TextUpdate(object sender, EventArgs e)
{ if(cmbGD.Items.Count>0)
    cmbGD.SelectedIndex = 0; }

private void button1_Click(object sender, EventArgs e)
{
    buscarPacienteS();
}

public void borrarNomNP()
{
    txtNom.Text = "";
    txtNP.Text = "";
    NP = 0;
    idP = 0;
}

```

```

}

public void buscarPacienteS()
{
    borrarInfo();
    borrarNomNP();
    frmBuscarPacienteS a = new frmBuscarPacienteS(true);
    a.ShowDialog();
    if (NP > 0 && idP > 0)
    {
        MySqlDataReader dr = frmLogin.con.seleccionarCatalogo("PACIENTES",
        "*", " WHERE IDPACIENTE=" + idP.ToString());
        if (dr.Read())
        {
            txtNP.Text= dr.GetString("NUMERO_PACIENTE");
            txtNom.Text = dr.GetString("NOMBRE") + " " +
            dr.GetString("APELLIDO_PATERNO") + " " + dr.GetString("APELLIDO_MATERNO");
            if (!dr.IsDBNull(8))
            {
                string idNiv = dr.GetString("IDNIVEL_SOCIOECONOMICO");
                string idViv = dr.GetString("IDVIVIENDA");
                string idGru = dr.GetString("IDGRUPO_ETNICO");
                string idRel = dr.GetString("IDRELIGION");
                string idSex = dr.GetString("IDSEXO");
                string idSan = dr.GetString("IDSANGRE");
                dr.Close();
                cmbNiv.Text =
                frmLogin.con.buscarElemento("CAT_NIVELES_SOCIOECONOMICOS",
                "NIVEL_SOCIOECONOMICO", " WHERE IDNIVEL_SOCIOECONOMICO=" + idNiv);
                cmbViv.Text = frmLogin.con.buscarElemento("CAT_VIVIENDAS",
                "VIVIENDA", " WHERE IDVIVIENDA=" + idViv);
                cmbGru.Text =
                frmLogin.con.buscarElemento("CAT_GRUPOS_ETNICOS", "GRUPO_ETNICO", "
                WHERE IDGRUPO_ETNICO=" + idGru);
                cmbRel.Text = frmLogin.con.buscarElemento("CAT_RELIGIONES",
                "RELIGION", " WHERE IDRELIGION=" + idRel);
            }
        }
    }
}

```



```
        cmbSex.Text = frmLogin.con.buscarElemento("CAT_SEXOS", "SEXO",  
" WHERE IDSEXO=" + idSex);  
        cmbSan.Text = frmLogin.con.buscarElemento("CAT_SANGRE",  
"SANGRE", " WHERE IDSANGRE=" + idSan);
```

```
        frmLogin.con.leerPacienteEDA(idP.ToString(), lstEnf2, lstDis2, lstAle2);
```

```
        if(lstAle1.Items.Count>0)  
            lstAle1.SelectedIndex = 0;  
        sel(gpoGE);  
        sel(gpoAle);  
        sel(gpoGD);  
        if(lstEnf2.Items.Count>0)  
            lstEnf2.SelectedIndex = 0;  
        if(lstDis2.Items.Count>0)  
            lstDis2.SelectedIndex = 0;  
        if(lstAle2.Items.Count>0)  
            lstAle2.SelectedIndex = 0;
```

```
    }  
    else  
    {  
        if (!dr.IsClosed)  
            dr.Close();  
        borrarInfo();  
    }
```

```
        if(!dr.IsClosed)  
            dr.Close();  
    }
```

```
    }  
}
```

```
private void frmDatosSanidad_Load(object sender, EventArgs e)  
{
```

```
        frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "4",  
"1", "I a DS -" + frmLogin.uLogeado.Usuario, frmLogin.fechaHoyMySQL(),  
frmLogin.horaMySQL());
```

```
        buscarPacienteS();  
    }
```

```
private void btnEnf1_Click(object sender, EventArgs e)  
{agregarArticulo(lstEnf1, lstEnf2);}
```

```
private void btnEnf2_Click(object sender, EventArgs e)  
{quitarArticulo(lstEnf2);}
```

```
private void btnDis1_Click(object sender, EventArgs e)  
{ agregarArticulo(lstDis1, lstDis2); }
```

```
private void btnDis2_Click(object sender, EventArgs e)  
{ quitarArticulo(lstDis2); }
```

```
private void btnAle1_Click(object sender, EventArgs e)  
{ agregarArticulo(lstAle1, lstAle2); }
```

```
private void btnAle2_Click(object sender, EventArgs e)  
{ quitarArticulo(lstAle2); }
```

```
public static void agregarArticulo(ListBox origen, ListBox destino)  
{  
    if (!destino.Items.Contains(origen.Text))  
    {  
        destino.Items.Add(origen.Text);  
        destino.SelectedIndex=destino.Items.Count - 1;  
        if (origen.SelectedIndex < origen.Items.Count - 1)  
            origen.SelectedIndex += 1;  
    }  
}
```

```
public static void quitarArticulo(ListBox origen)  
{
```

```

if (origen.Items.Count > 0)
{
    int i = origen.SelectedIndex;
    origen.Items.RemoveAt(origen.SelectedIndex);
    origen.SelectedIndex = i - 1;
    if(i==0 && origen.Items.Count>0)
        origen.SelectedIndex = 0;
}
}

private void cmbGru_TextUpdate(object sender, EventArgs e)
{ if(cmbGru.Items.Count>0)
    cmbGru.SelectedIndex = 0; }

private void frmDatosSanidad_FormClosing(object sender,
FormClosingEventArgs e)
{
    frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "4",
"2", "S de DS -" + frmLogin.uLogeado.Usuario, frmLogin.fechaHoyMySQL(),
frmLogin.horaMySQL());
}
}
}

```

"frmLogin.cs"

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Diagnostics; // Process
using MySql.Data.MySqlClient; // dataReader

namespace Health_Card_Manager
{
    public partial class frmLogin : Form
    {
        public static conexionMySQL con = new conexionMySQL();
        public static cUsuario uLogeado = new cUsuario();

        public frmLogin()
        {
            InitializeComponent();
        }

        private void frmLogin_FormClosing(object sender, FormClosingEventArgs e)
        {
            Application.Exit();
        }

        public static void cerrarr(FormClosingEventArgs e)
        {
            DialogResult res = MessageBox.Show("¿Desea salir de la aplicación?", "Health
Card Manager 2010", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (res == DialogResult.Yes)
            {
```

```

        frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "1",
"2", "S de S -" + frmLogin.uLogeado.Usuario, frmLogin.fechaHoyMySQL(),
frmLogin.horaMySQL());
        Process a = Process.GetCurrentProcess();
        a.Kill();
    }
    else if(e is FormClosingEventArgs)
    {
        e.Cancel = true;
    }
}

private void btnEnt_Click(object sender, EventArgs e)
{
    validarUsuario();
}

public void validarUsuario()
{
    if (validarCampos(txtUsu.Text, txtCon.Text))
        validarUsuarioDB(txtUsu.Text, txtCon.Text);
}

public bool validarCampos(string usuario, string contraseña)
{
    if (usuario == "")
        MessageBox.Show("Ingresar Usuario.", "Iniciar Sesión",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    else if (contraseña == "")
        MessageBox.Show("Ingresar Constraseña.", "Iniciar Sesión",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    else
        return true;
    return false;
}

private void validarUsuarioDB(string usuario, string contraseña)

```

```

{
    if (con.conectar())
    {
        MySqlDataReader dr = con.seleccionarCatalogo("USUARIOS", "*", "
WHERE USUARIO=" + usuario + "");
        if (dr.Read())
        {
            if (dr.GetString("CONTRASENA").CompareTo(contrasena) == 0)
            {
                if
(DateTime.Compare(Convert.ToDateTime(dr.GetString("FECHA_BAJA")),
DateTime.Now) > 0)
                {
                    if (dr.GetBoolean("ACTIVADO"))
                    {
                        uLogeado.BDusuarioTOcUsuario(dr);
                        this.Hide();
                        frmPrincipal a = new frmPrincipal();
                        con.insertarAuditoria(uLogeado.IdUsuario.ToString(), "1", "1", "I al
S -" + uLogeado.Usuario, fechaHoyMySQL(), horaMySQL());
                        a.Show();
                    }
                    else
                        MessageBox.Show("El usuario " + usuario + " esta desactivado
del sistema.", "Iniciar Sesión", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
                else
                    MessageBox.Show("El usuario " + usuario + " fue dado de baja.",
"Iniciar Sesión", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            else
                MessageBox.Show("Contraseña invalida.", "Iniciar Sesión",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else
            MessageBox.Show("No existe el usuario " + usuario + ".", "Iniciar
Sesión", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```

    }
}
public static string fechaHoyMySQL()
{
    DateTime f = DateTime.Now;
    return f.Year.ToString() + "-" + f.Month.ToString() + "-" + f.Day.ToString();
}
public static string horaMySQL()
{
    DateTime f = DateTime.Now; ;
    return f.Hour.ToString() + ":" + f.Minute.ToString() + ":" + f.Second.ToString();
}
public static string convetirFechaMySql(string f)
{
    return f.Split('/')[2].Substring(0,4) + "-" + f.Split('/')[1] + "-" + f.Split('/')[0] ;
}
public int comparaFechasMySQL(string fecha1, string fecha2)
{
    string[] fs1=fecha1.Split('-');
    MessageBox.Show(fecha1 + " " + fs1.Length.ToString());
    int f1 = int.Parse(fs1[0].ToString() + fs1[1].ToString() + fs1[2].ToString());
    string[] fs2 = fecha2.Split('-');
    int f2 = int.Parse(fs2[0].ToString() + fs2[1].ToString() + fs2[2].ToString());
    if (f1 > f2)
        return 1; //Si la fecha1 es mayor a la fecha2
    if (f1 < f2)
        return -1; //Si la fecha1 es menor a la fecha2
    return 0; //Si la fecha1 es igual a la fecha2
}

private void textCon_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == Convert.ToChar(Keys.Return))
    { validarUsuario();}
}
}
}

```

"frmPresentacion.cs"

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Diagnostics;

namespace Health_Card_Manager
{
    public partial class frmPresentacion : Form
    {

        public frmPresentacion()
        {
            InitializeComponent();
        }

        private void frmPresentacion_Load(object sender, EventArgs e)
        {
            Process a = Process.GetCurrentProcess();
            Process[] procesos = Process.GetProcessesByName(a.ProcessName);
            if (procesos.Length > 1)
            {
                MessageBox.Show("Ya se encuentra en ejecución", "Health Card Manager
2010", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                Process p = Process.GetCurrentProcess();
                p.Kill();
            }
            else
            {
                conexionMySql con = new conexionMySql();
                if (con.conectar() == true)
```



```
        tiempo.Start();
    else
    {
        Process p = Process.GetCurrentProcess();
        p.Kill();
    }
}

private void tiempo_Tick(object sender, EventArgs e)
{
    this.Hide();
    frmLogin frmLo = new frmLogin();
    frmLo.Show();
    tiempo.Stop();
}
}
```

“frmPrincipal.cs”

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Diagnostics;

namespace Health_Card_Manager
{
    public partial class frmPrincipal : Form
    {
        public frmPrincipal()
        {
            InitializeComponent();
            inicializaMenuV();
        }

        private void frmMenu_FormClosing(object sender, FormClosingEventArgs e)
        {
            frmLogin.cerrarr(e);
        }

        private void salirToolStripMenuItem_Click(object sender, EventArgs e)
        {
            frmMenu_FormClosing(null, null);
        }

        public void inicializaMenuV()
        {
            if (frmLogin.con.conectar())
            {
                MySql.Data.MySqlClient.MySqlDataReader dr =
                frmLogin.con.obtenerModulos(frmLogin.uLogeado.IdUsuario);
            }
        }
    }
}
```

```

while (dr.Read())
{
    foreach (ToolStripItem c in menuVer.Items)
        if (c.Visible == false && dr.GetString("MODULO").CompareTo(c.Tag) ==
0)
            c.Visible = true;
    foreach (ToolStripItem c in menuVer2.Items)
        if (c.Visible == false && dr.GetString("MODULO").CompareTo(c.Tag) ==
0)
            c.Visible = true;
}
}
}

```

```

private void cambiarContrasenaToolStripMenuItem_Click(object sender,
EventArgs e)

```

```

{
    frmContrasena a = new frmContrasena();
    formAbierto(a);
}

```

```

private void btnCU_Click(object sender, EventArgs e)

```

```

{
    frmControlUsuarios a = new frmControlUsuarios();
    formAbierto(a);
}

```

```

public void formAbierto(Form formulario)

```

```

{
    foreach (Form f in Application.OpenForms)
        if (f.Tag == formulario.Tag)
        {
            MessageBox.Show(""" + formulario.Text + "" ya se encuentra
abierto.", "Health Card Manager
2010", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            formulario.Dispose();
            return;
        }
}

```

```
        formulario.Show();
    }

    private void btnCP_Click(object sender, EventArgs e)
    {
        frmControlPacientes a = new frmControlPacientes();
        formAbierto(a);
    }

    private void btnDS_Click(object sender, EventArgs e)
    {
        frmDatosSanidad a = new frmDatosSanidad();
        formAbierto(a);
    }

    private void btnC_Click(object sender, EventArgs e)
    {
        frmCredencializacion a = new frmCredencializacion();
        formAbierto(a);
    }

    private void btnR_Click(object sender, EventArgs e)
    {
        frmRecoleccion a = new frmRecoleccion();
        formAbierto(a);
    }

    private void acToolStripMenuItem_Click(object sender, EventArgs e)
    {
        frmAcerda a = new frmAcerda();
        a.ShowDialog();
    }
}
}
```

"frmRecoleccion.cs"

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Health_Card_Manager
{
    public partial class frmRecoleccion : Form
    {
        public frmRecoleccion()
        {
            InitializeComponent();
        }

        private void btnAce_Click(object sender, EventArgs e)
        {
            iniciarBusqueda();
        }

        public void iniciarBusqueda()
        {
            try
            {
                borrarInfo();
                Queue<string> elementos = new Queue<string>();
                string SN = "";
                int res = ACR120U.ACR120_Open(0);
                if (res == 0)
                {
                    long sto = 0;
                    byte[] ResultSN = new byte[11];
                    byte LengthTag = 0x00; ;
                    byte[] TagType = new byte[51];
```

```

        res = ACR120U.ACR120_Select(0, ref TagType[0], ref LengthTag, ref
ResultSN[0]);
        if (res == 62536)
        {
            MessageBox.Show("Colocar credencial sobre el lector ACR120U",
"Seleccionar credencial", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }
        else if (res < 0)
        {
            MessageBox.Show(ACR120U.GetErrMsg(res), "Seleccionar
credencial", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }
        else
        {

            string tipoCredencial = ACR120U.GetTagType1(TagType[0]);
            if (tipoCredencial.CompareTo("Mifare 4K") == 0 ||
tipoCredencial.CompareTo("Mifare 1K") == 0)
            {

                for (int ctr = 0; ctr < LengthTag; ctr++)
                    SN = SN + string.Format("{0:X2} ", ResultSN[ctr]);
                byte[] key = { 255, 255, 255, 255, 255, 255 };
                int sector;
                for (sector = 0; sector < 39; sector++)//32 => 62506
                {

                    res = ACR120U.ACR120_Login(0, Convert.ToByte(sector),
Convert.ToInt16(ACR120U.ACR120_LOGIN_KEYTYPE_A), Convert.ToByte(sto), ref
key[0]);

                    if (res != 0)
                        break;
                }
            }
        }

```

```

        if (res < 0)
            MessageBox.Show(ACR120U.GetErrMsg(res), "Verificando clave
del sector " + sector.ToString());
        else if (res > 0)
        {
            if (sector == 0)
                MessageBox.Show(res.ToString() + ": La clave del sector no
corresponde", "Verificando clave del sector " + sector.ToString());
            else
                MessageBox.Show(res.ToString() + ": La credencial "" + SN + ""
fue retirada del \n lector o la clave del lector no corresponde", "Verificando clave del
sector " + sector.ToString());
        }
        else
        {
            int resW = 0;
            int modulo = 0;
            byte[] ele = new byte[16];
            for (sector = 1; sector < 39; sector++)
            {
                res = ACR120U.ACR120_Login(0, Convert.ToByte(sector),
Convert.ToInt16(ACR120U.ACR120_LOGIN_KEYTYPE_A), Convert.ToByte(sto), ref
key[0]);

                if (res != 0)
                {
                    if (res < 0)
                        MessageBox.Show(ACR120U.GetErrMsg(res), "Validando
clave del sector " + sector.ToString());
                    else if (res > 0)
                    {
                        if (sector == 1)
                            MessageBox.Show(res.ToString() + ": La clave del
sector no corresponde", "Verificando clave del sector " + sector.ToString());
                        else

```





```

else
    MessageBox.Show("Tarjetas " + tipoCredencial + " no son
soportadas por la aplicacion", "Tarjeta Mifare Classic 4K", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
}
Fin:
string sanidad = "";
while (elementos.Count > 0)
sanidad += elementos.Dequeue();
string[] elem = sanidad.Split('$');
ACR120U.ACR120_Close(0);
if (elem.Length > 11)
{
    lbINP.Text = elem[0];
    lbINom.Text = elem[1];
    lbISex.Text = elem[2];
    lbISan.Text = elem[3];
    lbINiv.Text = elem[4];
    lbIViv.Text = elem[5];
    lbIGru.Text = elem[6];
    lbIRel.Text = elem[7];
    if (elem[8].Split(',').Length > 0)
        for (int j = 0; j < elem[8].Split(',').Length; j++)
            lstEnf2.Items.Add(elem[8].Split(',')[j]);
    if (elem[9].Split(',').Length > 0)
        for (int j = 0; j < elem[9].Split(',').Length; j++)
            lstDis2.Items.Add(elem[9].Split(',')[j]);
    if (elem[10].Split(',').Length > 0)
        for (int j = 0; j < elem[10].Split(',').Length; j++)
            lstAle2.Items.Add(elem[10].Split(',')[j]);
    lbITel.Text = elem[11];
    lbINC.Text = SN;
    frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(),
"6", "5", "M " + SN + "leida -" + frmLogin.uLogeado.Usuario,
frmLogin.fechaHoyMySQL(), frmLogin.horaMySQL());
}
}
}

```

```

        else
            MessageBox.Show(ACR120U.GetErrMsg(res), "Conectar el lector
ACR120U", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error de credencializacion",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```

public void borrarInfo()

```

```

{
    lstAle2.Items.Clear();
    lstDis2.Items.Clear();
    lstEnf2.Items.Clear();
    lbINom.Text = "";
    lbINP.Text = "";
    lbINC.Text = "";
    lbISex.Text = "";
    lbISan.Text = "";
    lbIViv.Text = "";
    lbINiv.Text = "";
    lbIRel.Text = "";
    lbIGru.Text = "";
    lbITel.Text = "";
}

```

```

public static string B2S(byte[] arreglo)

```

```

{
    byte[] dBytes = arreglo;
    string str;
    System.Text.UTF8Encoding enc = new System.Text.UTF8Encoding();

```

```

        str = enc.GetString(dBytes);
        return str;
    }

    private void frmRecoleccion_Load(object sender, EventArgs e)
    {
        frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "6",
"1", "I a R -" + frmLogin.uLogeado.Usuario, frmLogin.fechaHoyMySQL(),
frmLogin.horaMySQL());
    }

    private void frmRecoleccion_FormClosing(object sender, FormClosingEventArgs
e)
    {
        frmLogin.con.insertarAuditoria(frmLogin.uLogeado.IdUsuario.ToString(), "6",
"2", "S de R -" + frmLogin.uLogeado.Usuario, frmLogin.fechaHoyMySQL(),
frmLogin.horaMySQL());
    }
}
}
}

```

## BIBLIOGRAFIA

[1] [http://www.osanet.euskadi.net/r85-](http://www.osanet.euskadi.net/r85-ona/es/contenidos/informacion/ona_txartela/es_ona/que_es_ona.html)

[ona/es/contenidos/informacion/ona\\_txartela/es\\_ona/que\\_es\\_ona.html](http://www.osanet.euskadi.net/r85-ona/es/contenidos/informacion/ona_txartela/es_ona/que_es_ona.html)

15.02.2010

[2] [http://es.wikipedia.org/wiki/Tarjeta\\_ONA](http://es.wikipedia.org/wiki/Tarjeta_ONA)

15.02.2010

[3] <http://ec.europa.eu/social/main.jsp?catId=559&langId=es>

17.02.2010

[4] [http://www.emexico.gob.mx/wb2/eMex/eMex\\_d37b\\_not123\\_tarjeta\\_de\\_se](http://www.emexico.gob.mx/wb2/eMex/eMex_d37b_not123_tarjeta_de_se)

17.02.2010

[5] <http://www.politicadigital.com.mx/?P=leernoticia&Article=746&c=104>

20.02.2010

[6] Periódico MILENIO, México, Blanca Valadez

Habrá credencial electrónica con historial clínico, Pagina 38

22.02.2010

[7] <http://www.salud.gob.mx/unidades/cdi/nom/168ssa18.html>

20.02.2010

[8] 2006, UAM Azcapotzalco. Ingeniería en Electrónica

“SISTEMA DE IDENTIFICACIÓN Y ACCESO MEDIANTE TARJETAS INTELIGENTES”

Autor: Reyes Boizo Gustavo Jhonatan, Asesor: Godínez Bravo Ricardo

16.03.2010



**Advanced Card Systems Ltd.**  
Card & Reader Technologies

# ACR120 Contactless Reader/Writer



Technical Specifications



## Table of Contents

<b>1.0.</b>	<b>Introduction .....</b>	<b>3</b>
<b>2.0.</b>	<b>Features .....</b>	<b>4</b>
<b>3.0.</b>	<b>Typical Applications.....</b>	<b>5</b>
<b>4.0.</b>	<b>Technical Specifications.....</b>	<b>6</b>
4.1.	Configuration of ACR120 reader .....	6
4.2.	Configuration of ACR120 reader module .....	7



## 1.0. Introduction



The ACR120 is a compact and cost-effective contactless reader and writer. It is developed based on the 13.56 MHz contactless smart card (RFID) technology, supporting Mifare and ISO 14443 Type A and B cards. Its proximity operating distance is up to 5 cm, depending on the type of contactless tag in use.

The versatile reader is available in both USB and Serial interface version, which can easily be integrated into a PC environment as well as other systems. It is ideal for a broad range of applications, including public transport terminals, physical and logical access controls, and even vending machines.

The operation of ACR120 is extremely quick and convenient. All read/write operations can be done by waving the contactless card near the device. Moreover, its compact design enables it to be mounted and located anywhere with ease. ACR120 is also available in module form, giving you maximum development flexibility.



## 2.0. Features

- USB PnP or serial RS232 interface (Also available in RS485 *upon request*)
- Read and write functionality
- Built-in antenna for contactless tag access, with card reading distance of up to 50 mm
- Supports major contactless smart cards conforming to ISO 14443 Type A and B in the market
- Supports Mifare cards, including new Mifare Ultralight C and Mifare Plus SL1
- Built-in anti-collision feature (at least 1 card is detected when multiple cards are presented)
- Selective card polling capability (especially useful when multiple cards are presented)
- Operation LED
- Buzzer
- OEM PCBA module version (*Upon request*)





### 3.0. Typical Applications

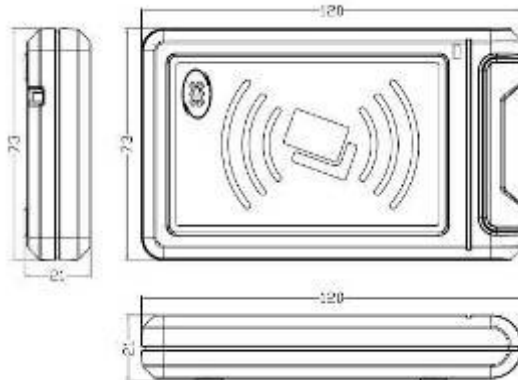


- Public transportation terminals
- Automatic fare collection
- Vending machines
- Physical access control
- Logical access control
- Time and attendance
- Contactless public phones
- Metering



## 4.0. Technical Specifications

### 4.1. Configuration of ACR120 reader



#### Serial Interface

Type ..... RS232 (Standard)  
 Operation Baud Rate ..... 9,600-115,200 bps  
 Power Source ..... From PS/2 mouse interface  
 Supply Voltage ..... Regulated 5 V DC (4.75 – 5.25V)  
 Supply Current ..... 200 mA

#### Universal Serial Bus Interface

Power source ..... From USB  
 Speed ..... 1.5 Mbps (Low Speed)  
 Supply Voltage ..... Regulated 5 V DC  
 Supply Current ..... 200 mA

#### Contactless Smart Card Interface

Standard ..... ISO 14443 A & B  
 Protocol ..... Mifare® Classic protocols, Mifare® DESFire protocols, Mifare Mini protocols  
 Smart card read / write speed ..... 106 kbps

#### Case

Dimensions ..... 120 mm (L) x 73 mm (W) x 20 mm (H)  
 Weight ..... 155 g (with cable)  
 Material ..... ABS  
 Color ..... Metallic Silver Grey ■

#### Antenna

Antenna Size ..... 55 mm x 85 mm  
 Operating distance ..... up to 50 mm

#### Operating Frequency

Operating Frequency ..... 13.56 MHz

#### Operating Conditions

Temperature ..... 0 - 50° C  
 Humidity ..... 10% - 80%

#### Cable Connector

Length ..... 1.5 m (USB/ Serial RS232)

#### Certifications/Compliance

CE, FCC, RoHS Compliant, OFTA

#### Device Driver Operating System Support

Windows® 98, ME, NT (Serial), 2000, Server 2003, XP, Vista, Server 2008, Server 2008 R2, 7  
 Linux







**(B) PIN Assignment (for Serial Communication Interface – Alternate Connector)**

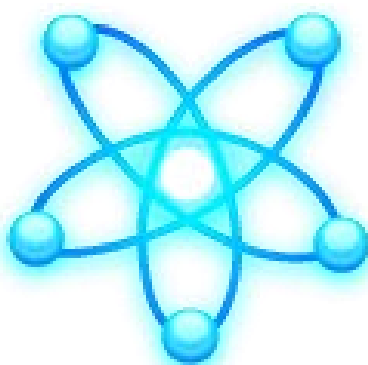
Pin	Signal	Description
1	RFU	Not connected
2	VCC	+5V supply to the module
3	GND	Power and signal Ground.
4	RS232Rx/RX485-	RS485-: (RS485 version) RS232 Receive (Rs232 version)
5	RS232Tx/RS485+	RS485+: (RS485 version) RS232 Transmit (RS232 version)
6	RFU	Not connected
7	RFU	Not connected

**(C) PIN Assignment (for USB Interface)**

Pin	Signal	Description
1	X	Not available
2	USB-	USB-: available
3	USB+	USB+: available
4	X	Not available
5	X	Not available
6	X	Not available
7	X	Not available
8	VCC	+5V supply to the module
9	X	Not available
10	X	Not available
11	X	Not available
12	GND	Power and signal Ground



# Health Card Manager 2010



## Manual de instalación



La instalación de Health Card Manager 2010 (HCM 2010) requiere la instalación previa de MySQL Server.

1. Abrir una línea de comando de MySQL Client y teclear el password del usuario root, como se muestra a continuación:

```
MySQL Command Line Client
Enter password: *****_
```

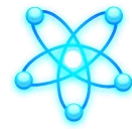
2. Creamos la base de datos “dbhealthcard2010” con el siguiente comando, como se muestra en la siguiente pantalla:

```
MySQL Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.45-community-nt MySQL Community Edition <GPL>
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

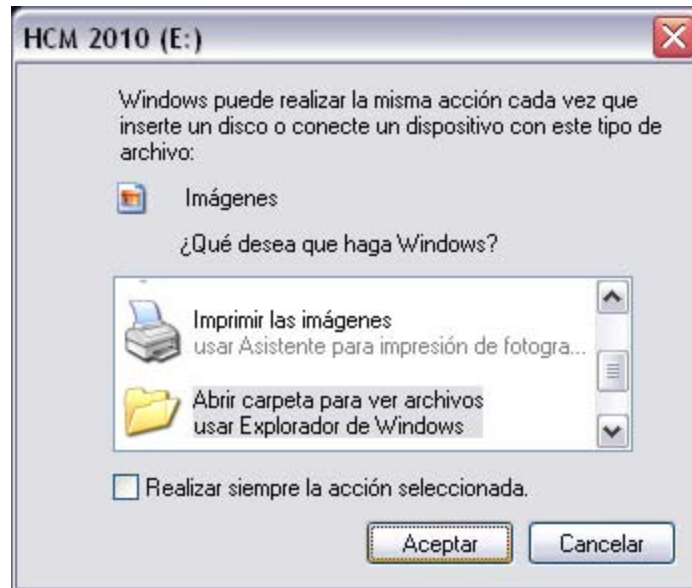
mysql> create database dbhealthcard2010;
Query OK, 1 row affected (0.00 sec)

mysql> _
```

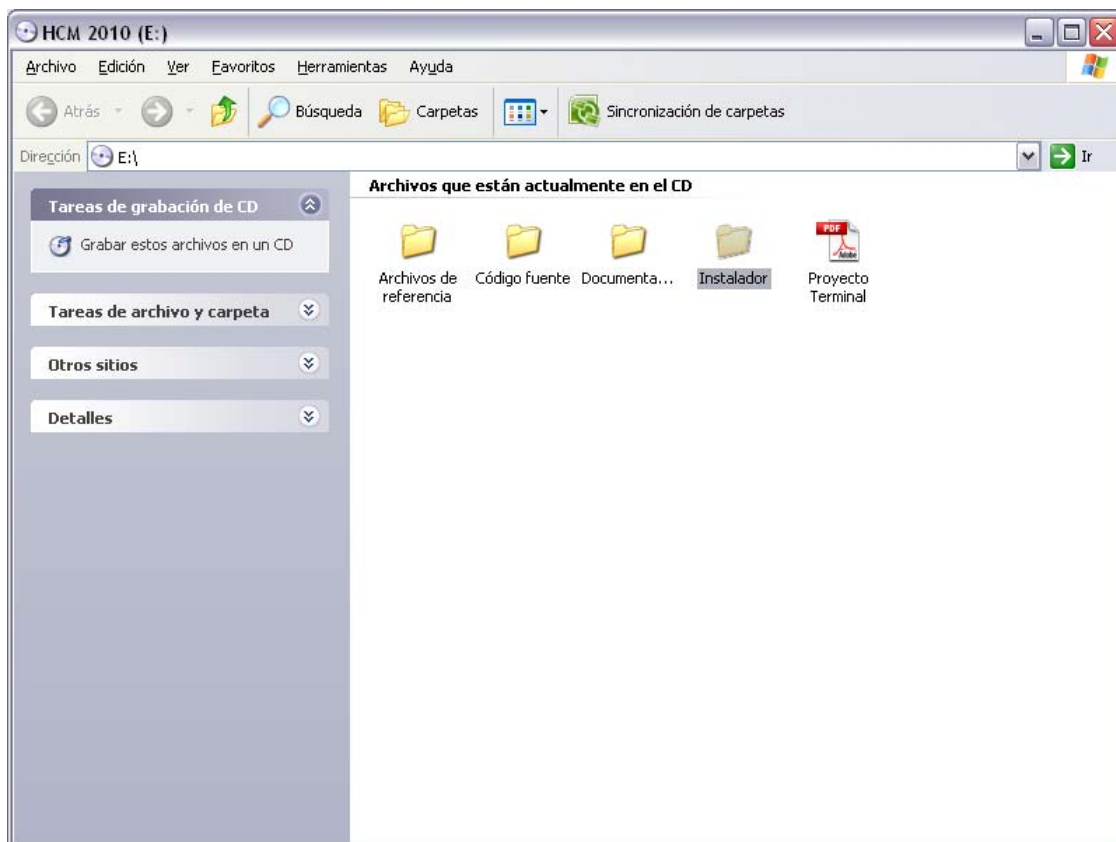
3. Cerramos la ventana de MySQL Client.



- Insertamos el CD (HCM 2010) en la unidad de disco, y se mostrará el siguiente cuadro de diálogo:

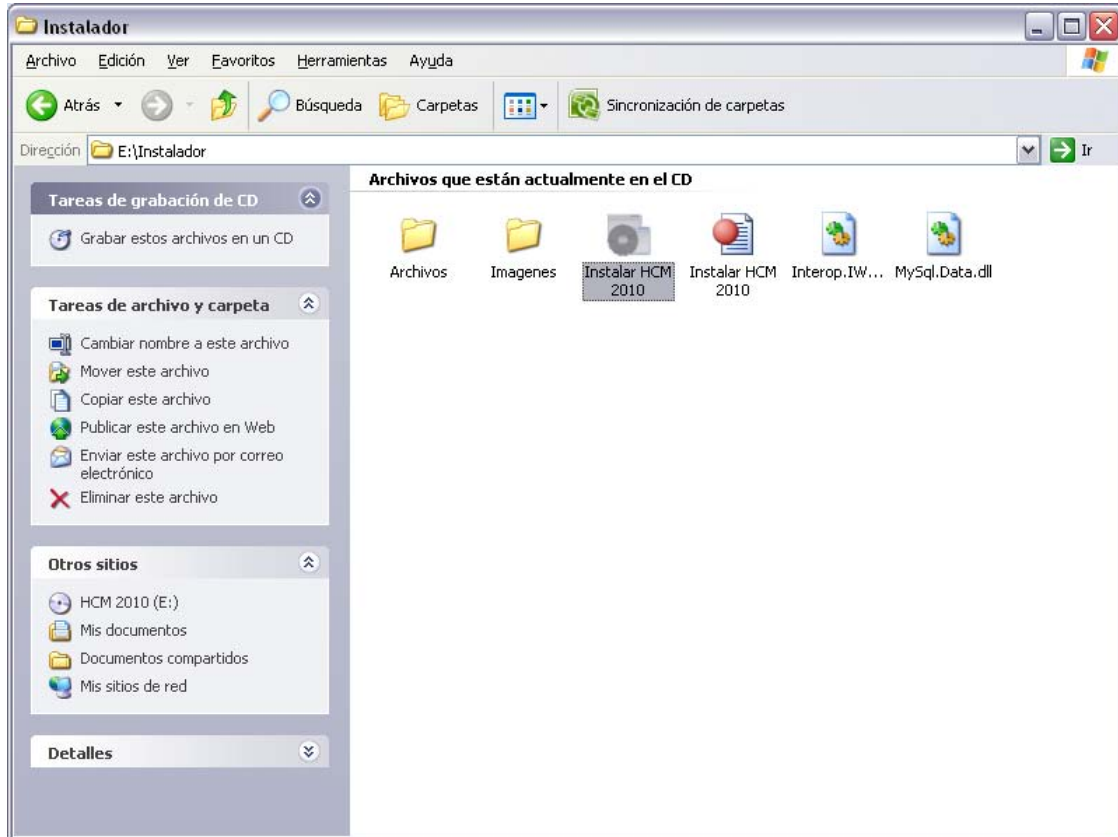
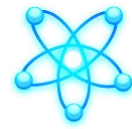


- Elegimos la opción “Abrir carpeta para ver archivos” y hacer clic en el botón “Aceptar”. Abrirá la ventana siguiente:



- Hacer doble clic sobre la carpeta “Instalador”, se abrirá la siguiente ventana:



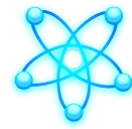


7. Hacer doble clic sobre el archivo ejecutable “Instalar HCM 2010” y se abrirá la siguiente ventana. Hacer clic sobre el botón “Crear” como se muestra a continuación:

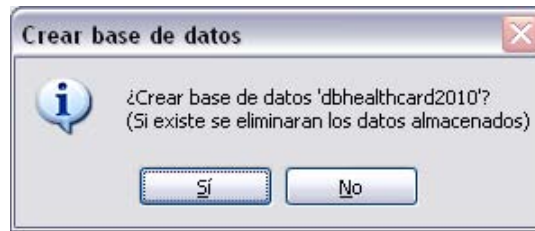


8. Se mostrará la siguiente ventana, ingresar el password y hacer clic a “Entrar”.

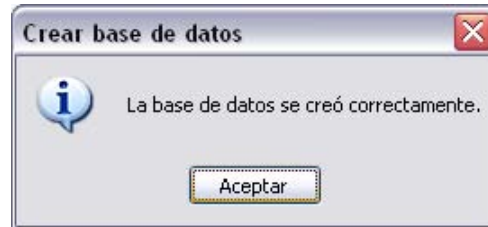




9. Se mostrará el siguiente cuadro de dialogo, hacer clic sobre el botón “Sí”



10. Aparecerá el siguiente mensaje.

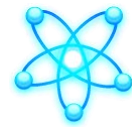


11. Hacer clic sobre el botón “Aceptar”, posteriormente hacer clic en el botón “Instalar” como se muestra a continuación:

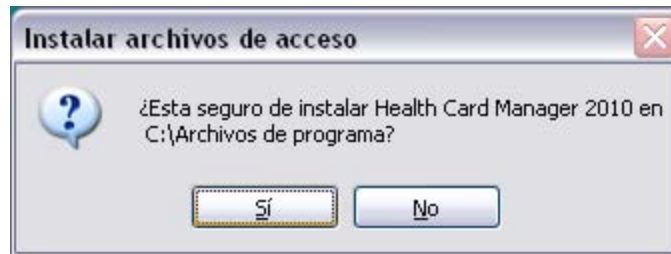


12. En el siguiente cuadro de dialogo elegir la carpeta donde desea instalar los archivos de acceso de HCM 2010.

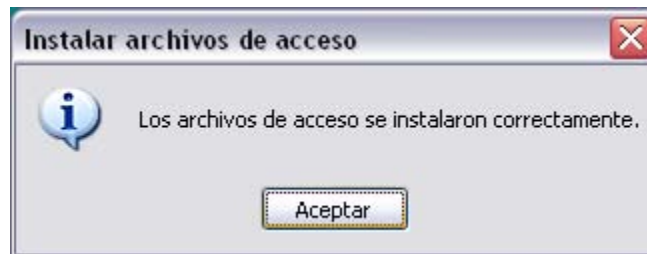




13. Se mostrara la siguiente ventana.



14. Hacer clic sobre el botón “Si”, se mostrará el siguiente mensaje:

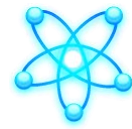


15. Hacer clic en el botón “Aceptar”. Se mostrará la siguiente ventana, hacer clic sobre el botón “Iniciar” como se muestra a continuación:



16. Aparecerá la ventana siguiente y hacer clic en “OK”



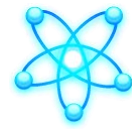


## 17. Preparando instalación...

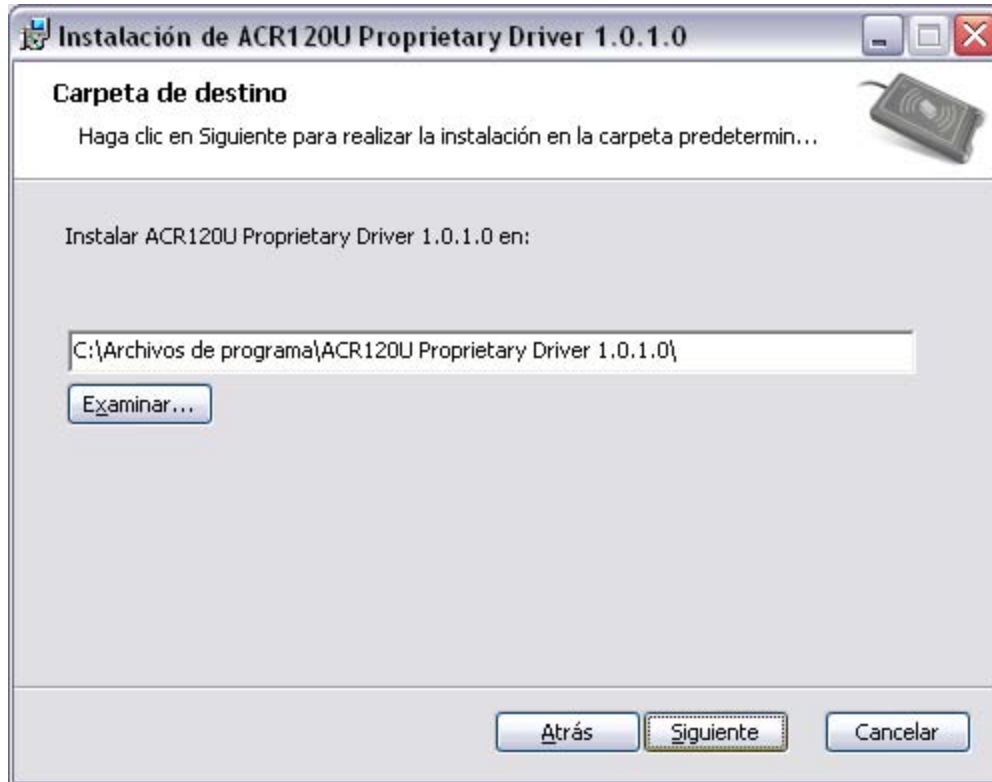


## 18. Aparecerá la ventana siguiente:



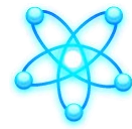


19. Elegir carpeta de destino y hacer clic en el botón “Siguiente”.

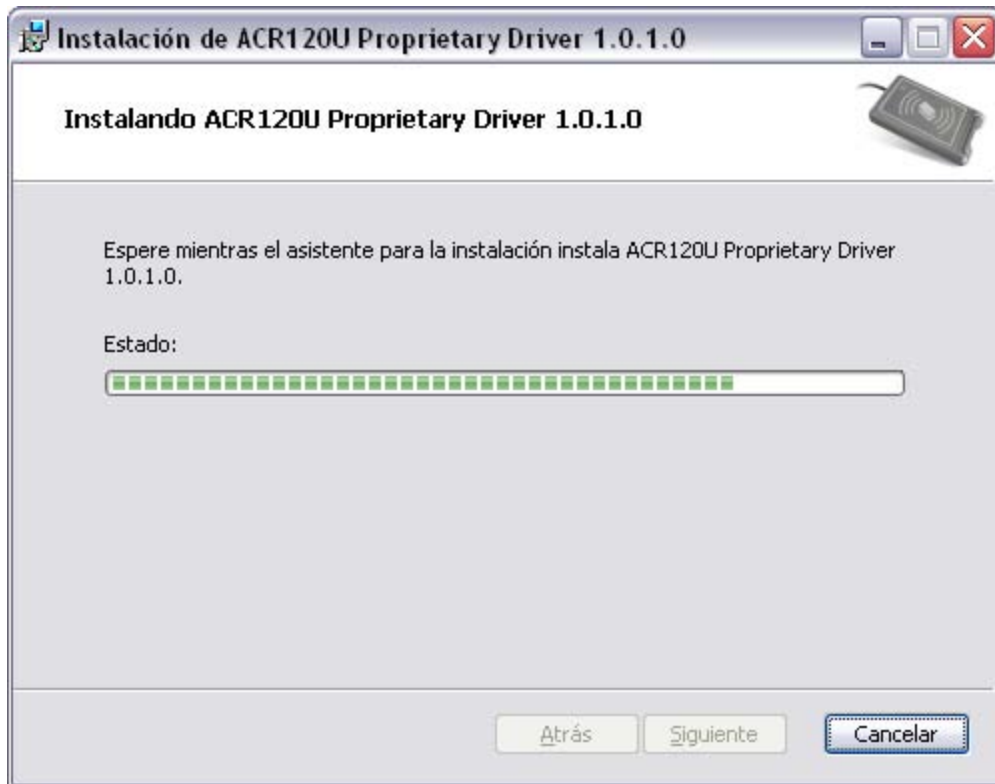


20. Hacer clic en el botón “Instalar” de la ventana siguiente:

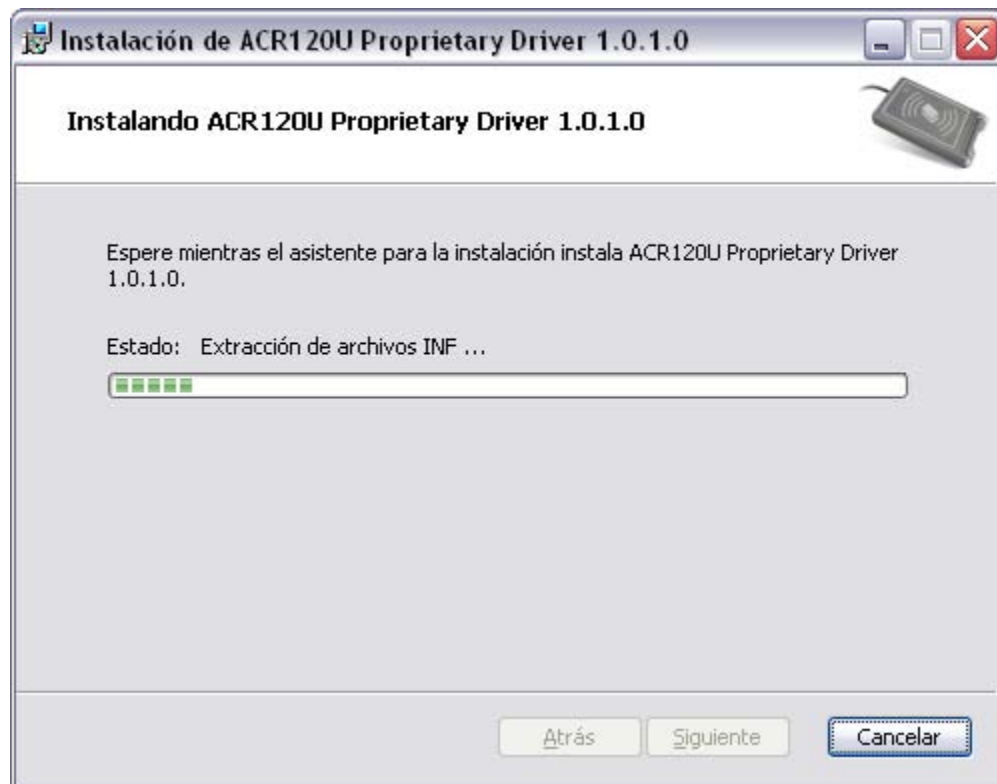




21. Esperar mientras termina la instalación en el cuadro siguiente:



22. Esperar mientras extrae archivos INF.





23. Hacer clic en botón “Finalizar”.



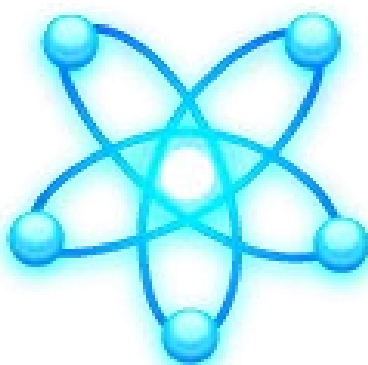
24. En el siguiente cuadro de dialogo, hacer clic en el botón “Aceptar”.



25. Cerrar la siguiente ventana. LISTO!



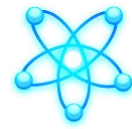
# Health Card Manager 2010



**Manual de usuario**

**Versión 1.0**





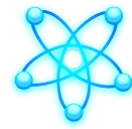
## 1. Ejecutar Health Card Manager 2010

Para ejecutar “Heath Card Manager 2010” (HCM 2010), hacemos doble clic sobre el icono de acceso directo que se instaló en nuestro escritorio como se muestra a continuación:



Posteriormente se mostrara la pantalla de presentación de HCM 2010.





## 2. Iniciar sesión

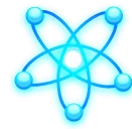
Para iniciar sesión en HCM 2010 se mostrará la siguiente pantalla, en la cual se tiene que ingresar un “Usuario” y una “Contraseña”. HCM 2010 crea un usuario con su contraseña por default, este usuario es “master0000” y su contraseña es de igual manera “master0000”.

Entonces por primera vez se tendrá que iniciar sesión con este usuario forzosamente.



Al realizar clic en el botón “entrar” se validara el usuario y mostrará la pantalla con los siguientes módulos.



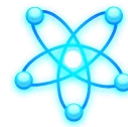


### 3. Control de usuarios

En el módulo “Control de usuarios” se pueden administrar todos los usuarios que van a tener acceso al sistema, en este módulo podemos dar de alta, baja y modificar un usuario. Al realizar clic sobre el botón “Control de usuarios”, como se muestra a continuación:



Se mostrará la siguiente pantalla:



#### 4. Alta de usuario

Para dar de alta un usuario se debe hacer clic sobre el botón de “alta” y posteriormente capturar los datos que se indican en la pantalla como se muestra a continuación:

Control de usuarios

Alta

Cédula: 2052021 CURP: GORA870218HMCDMN08

Nombre (s): ANGEL AKBAL

Apellido paterno: GODINEZ

Apellido materno: RAMIREZ

Especialidad: MEDICINA GENERAL

Subespecialidad: PEDIATRÍA

Rol: MASTER CLUES: DIF

Módulos:

- Control de usuarios
- Control de pacientes
- Datos de sanidad
- Credencialización
- Recolección

Usuario: angel0218

Contraseña: angel0218

Generar

Aceptar Cancelar

Una vez capturados los datos del usuario hacemos clic en el botón “Aceptar” y se desplegará el siguiente mensaje, en el cual seleccionaremos el botón de “Sí”.

Alta de usuario

¿Desea dar de alta al usuario 'angel0218'?

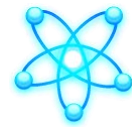
Sí No

Si el usuario se dio de alta correctamente aparecerá el siguiente mensaje. El cual aceptaremos.


Alta de usuario

El usuario 'angel0218' fue agregado exitosamente.

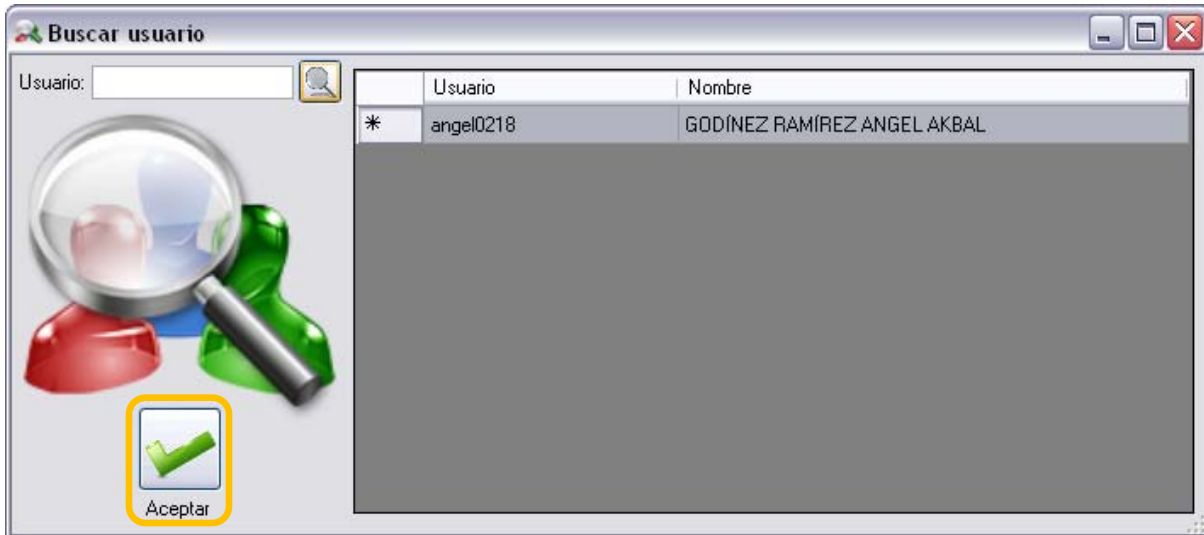
Aceptar



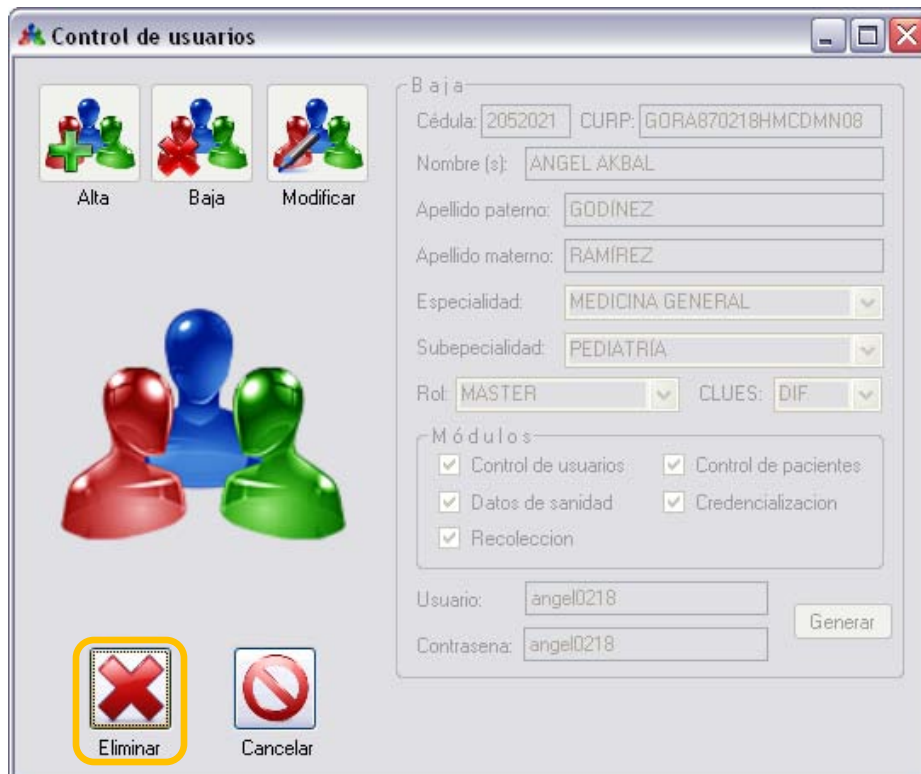
## 5. Baja de usuario

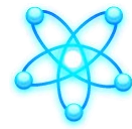
Para dar de baja un usuario se debe realizar clic sobre el botón de “baja”, mostrara la siguiente pantalla y al dar clic sobre el botón  desplegara todos los usuarios registrados y que no están dados de baja.

Se deberá elegir el usuario que se desea dar de baja y seleccionar el botón “Aceptar”.

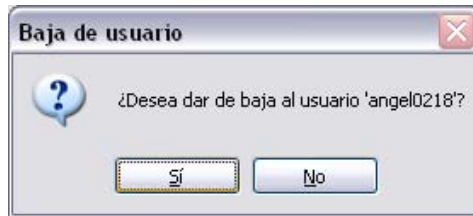


Posteriormente aparecerá toda la información del usuario que se desea dar de baja como se muestra a continuación:

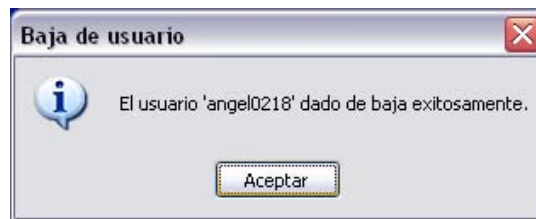





Una vez mostrada esta pantalla deberemos seleccionar el botón “Eliminar”, y se desplegará el siguiente mensaje, en el cual seleccionaremos el botón de “Si”.



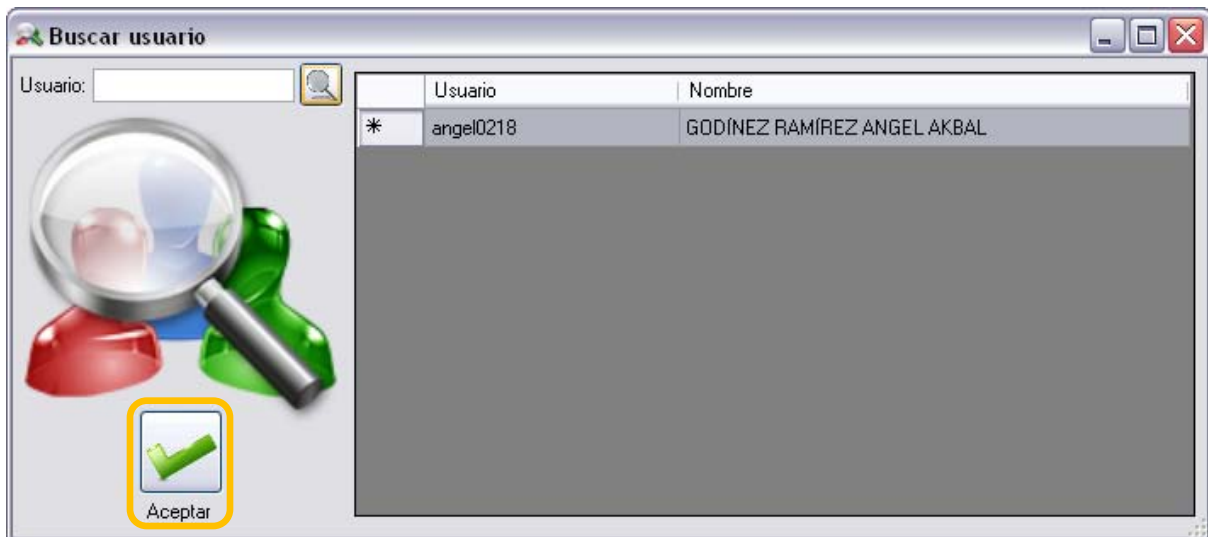
Si el usuario se dio de baja correctamente aparecerá el siguiente mensaje. El cual aceptaremos.

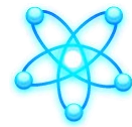


## 6. Modificar usuario

Para modificar un usuario se debe hacer clic sobre el botón de “modificar”, mostrará la siguiente pantalla y al dar clic sobre el botón  desplegará todos los usuarios registrados, aun los dados de baja.

Se deberá elegir el usuario que se desea modificar y seleccionar el botón “Aceptar”.





Posteriormente aparecerá toda la información del usuario que se desea modificar como se muestra a continuación:

**Control de usuarios**

**Modificar**

Cédula: 2052021 CURP: GORA870218HMCDMN08

Nombre (s): ANGEL AKBAL

Apellido paterno: GODINEZ

Apellido materno: RAMÍREZ

Especialidad: MEDICINA GENERAL

Subespecialidad: PEDIATRÍA

Rol: MASTER CLUES: DIF

Módulos:

- Control de usuarios
- Control de pacientes
- Datos de sanidad
- Credencialización
- Recoleccion

Usuario: angel0218 **Generar**

Contraseña: angel0218

Fecha de alta: 2011-01-12 Fecha de baja: 2012-01-12

Usuario activado.

**Aceptar** **Cancelar**

Una vez mostrada esta pantalla deberemos realizar los cambio pertinentes y seleccionar el botón “Aceptar”, y se desplegara el siguiente mensaje, en el cual seleccionaremos el botón de “Sí”.

**Modificar Usuario**

¿Desea guardar los cambios realizados?

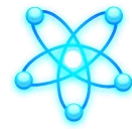
**Sí** **No**

Si el usuario se modifiko correctamente aparecerá el siguiente mensaje. El cual aceptaremos.

**Modificar Usuario**

El usuario 'angel0218' fue actualizado exitosamente.

**Aceptar**



## 7. Control de pacientes

En el módulo “Control de pacientes” se pueden administrar todos los pacientes que van a registrarse en el sistema, en este módulo podemos dar de alta, baja y modificar un paciente. Al realizar clic sobre el botón “Control de pacientes”, como se muestra a continuación:



Se mostrará la siguiente pantalla:

**Control de pacientes**

Alta Baja Modificar

Numero de paciente:

Nombre (s):

Apellido paterno:

Apellido materno:

CURP:

Fecha de nacimiento:

Telefono particular:

Tipo de domicilio:

Calle:  Número exterior:

Número interior:  Colonia:

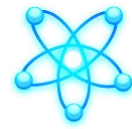
Estado:  Telefono de casa:

Municipio:  Código Postal:

Fecha de alta:  Fecha de baja:

Aceptar Cancelar





## 8. Alta de paciente

Para dar de alta un paciente se debe realizar clic sobre el botón de “alta” y posteriormente capturar los datos que se indican en la pantalla como se muestra a continuación:

**Control de pacientes**

**Alta**

Numero de paciente: 1101200957

Nombre (s): EZEQUIEL

Apellido paterno: CARRIZOZA

Apellido materno: REYNA

CURP: CARE751230JGLRMC53

Fecha de nacimiento: 12 Diciembre 1975

Telefono particular: 55-34-98-06-45

Tipo de domicilio: TRABAJO

Calle: RIO BALZAS Número exterior: 36

Número interior: Colonia: PORTEZUELOS

Estado: ESTADO DE MÉXICO Telefono de casa: 10-45-09-32

Municipio: CHIMALHUACÁN Código Postal: 40281

Fecha de alta: 2011-1-12 Fecha de baja: 2014-09-31

Botones: Alta, Baja, Modificar, Aceptar, Cancelar

Una vez capturados los datos del paciente hacemos clic en el botón “Aceptar” y se desplegará el siguiente mensaje, en el cual seleccionaremos el botón de “Si”.

**Alta de paciente**

¿Desea dar de alta el numero de paciente '1101200957'?

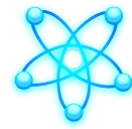
Botones: Sí, No

Si el paciente se dio de alta correctamente aparecerá el siguiente mensaje. El cual aceptaremos.


**Alta de paciente**

El paciente '1101200957' fue agregado exitosamente.

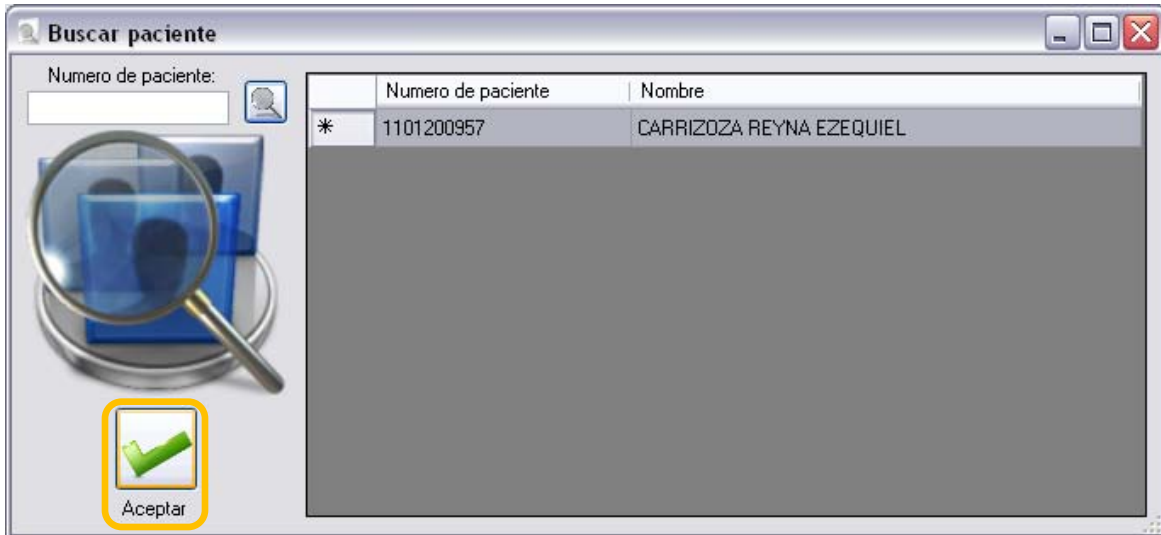
Botón: Aceptar



## 9. Baja de paciente

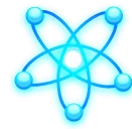
Para dar de baja un paciente se debe hacer clic sobre el botón de “baja”, mostrara la siguiente pantalla y al dar clic sobre el botón  desplegará todos los pacientes registrados y que no están dados de baja.

Se deberá elegir el paciente que se desea dar de baja y seleccionar el botón “Aceptar”.

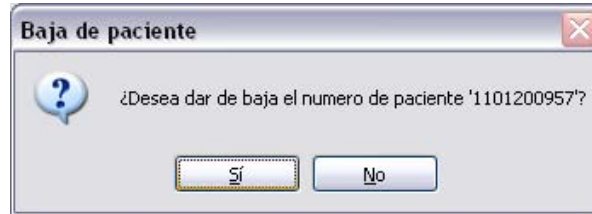


Posteriormente aparecerá toda la información del paciente que se desea dar de baja como se muestra a continuación:






Una vez mostrada esta pantalla deberemos seleccionar el botón “Eliminar”, y se desplegará el siguiente mensaje, en el cual seleccionaremos el botón de “Si”.



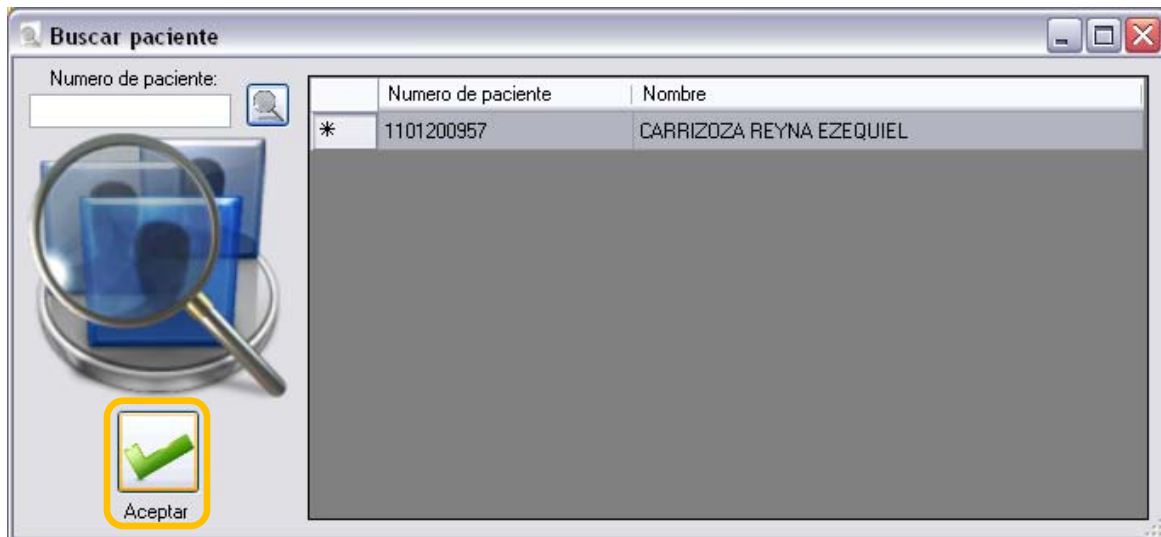
Si el paciente se dio de baja correctamente aparecerá el siguiente mensaje. El cual aceptaremos.

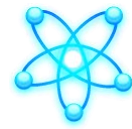


## 10. Modificar paciente

Para modificar un paciente se debe hacer clic sobre el botón de “modificar”, mostrará la siguiente pantalla y al dar clic sobre el botón  desplegará todos los pacientes registrados, aun los dados de baja.

Se deberá elegir el paciente que se desea modificar y seleccionar el botón “Aceptar”.





Posteriormente aparecerá toda la información del paciente que se desea modificar como se muestra a continuación:

The screenshot shows a window titled "Control de pacientes" with a "Modificar" tab. The form contains the following fields:

- Numero de paciente: 1101200957
- Nombre (s): EZEQUIEL
- Apellido paterno: CARRIZOZA
- Apellido materno: REYNA
- CURP: CARE751230UGLRMC53
- Fecha de nacimiento: 12 Diciembre 1975
- Telefono particular: 55-34-98-06-45
- Tipo de domicilio: TRABAJO
- Calle: RIO BALZAS, Número exterior: 36
- Número interior: (empty), Colonia: PORTEZUELOS
- Estado: ESTADO DE MÉXICO, Telefono de casa: 10-45-09-32
- Municipio: CHIMALHUACÁN, Código Postal: 40281
- Fecha de alta: 2011-01-12, Fecha de baja: 2015-01-12

Buttons: Alta, Baja, Modificar, Aceptar, Cancelar.

Una vez mostrada esta pantalla deberemos realizar los cambio pertinentes y seleccionar el botón "Aceptar", y se desplegara el siguiente mensaje, en el cual seleccionaremos el botón de "Sí".

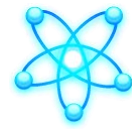
¿Desea guardar los cambios realizados?

Sí No

Si el paciente se modifico correctamente aparecerá el siguiente mensaje. El cual aceptaremos.

El paciente '1101200957' fue modificado exitosamente.



Aceptar

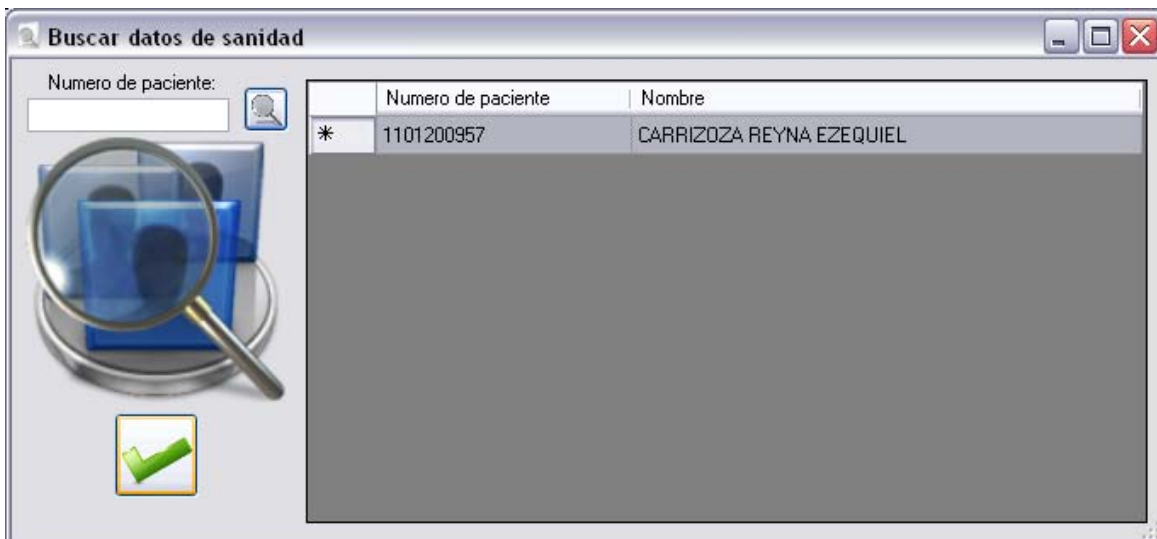


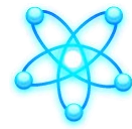
## 11. Datos de sanidad

En el módulo “Datos de Sanidad” se pueden administrar todos los datos de salud de los pacientes, en este módulo podemos verificar y modificar los datos de salud de un paciente. Al realizar clic sobre el botón “Datos de sanidad”, como se muestra a continuación:



Se mostrará la siguiente pantalla y al dar clic sobre el botón  desplegará todos los pacientes registrados. Se deberá elegir el paciente que se desea  modificar los datos de salud y seleccionar el botón “Aceptar”.





Posteriormente aparecerá toda la información de salud del paciente que se desea modificar como se muestra a continuación:

The screenshot shows a window titled "Datos de sanidad" with the following fields and options:

- N° de paciente: 1101200957 | Nombre: EZEQUIEL CARRIZOZA REYNA
- Nivel socioeconómico: C+ CLASE MEDIA ALTA | Sexo: MASCULINO
- Vivienda: CASA DE LADRILLO | Sangre: O+
- Grupo Etnico: NINGUNO | Religión: CATÓLICA
- Grupos de enfermedades: SISTEMA DIGESTIVO
- ENFERMEDAD DE CROHN
- PANCREATITIS
- SÍNDROME DE INTESTINO IRRITABLE
- REFLUJO GASTROESOFÁGICO
- Grupo de discapacidades: VISUAL
- AMBLIOPIA
- CEGUERA
- GLAUCOMA
- HEMIANOPSIA
- Alergias: AL HUEVO Y LA LECHE, MARISCOS Y PESCADOS, CACAHUATES, CHOCOLATE

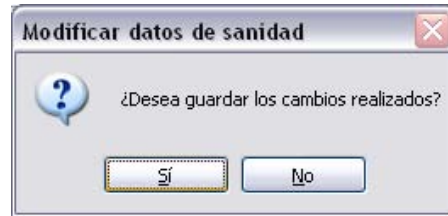
Una vez mostrada esta pantalla deberemos realizar los cambios pertinentes de la información de salud y seleccionar el botón "Aceptar".

The screenshot shows the same window with updated information:

- N° de paciente: 1101200957 | Nombre: EZEQUIEL CARRIZOZA REYNA
- Nivel socioeconómico: C+ CLASE MEDIA ALTA | Sexo: MASCULINO
- Vivienda: CASA DE LADRILLO | Sangre: O+
- Grupo Etnico: NINGUNO | Religión: CATÓLICA
- Grupos de enfermedades: TRASTORNOS MENTALES Y DEL COMPORTAMIENT
- DEPRESIÓN
- FIBROMIALGIA
- TRANSTORNOS DEL SUEÑO
- PANCREATITIS
- TRANSTORNOS DEL SUEÑO
- Grupo de discapacidades: FÍSICA
- ATAXIAS
- ATAXIA DE FRIEDREICH
- DISTONIA
- DISTROFIA MUSCULAR
- CEGUERA
- DISTONIA
- Alergias: SULFAMIDAS, ANTICONVULSIVOS, PENICILINA, POLVO
- ANTIBIÓTICOS
- PENICILINA



Se desplegará el siguiente mensaje, en el cual seleccionaremos el botón de “Sí”.



Si los datos de salud del paciente se modificaron correctamente aparecerá el siguiente mensaje. El cual aceptaremos.




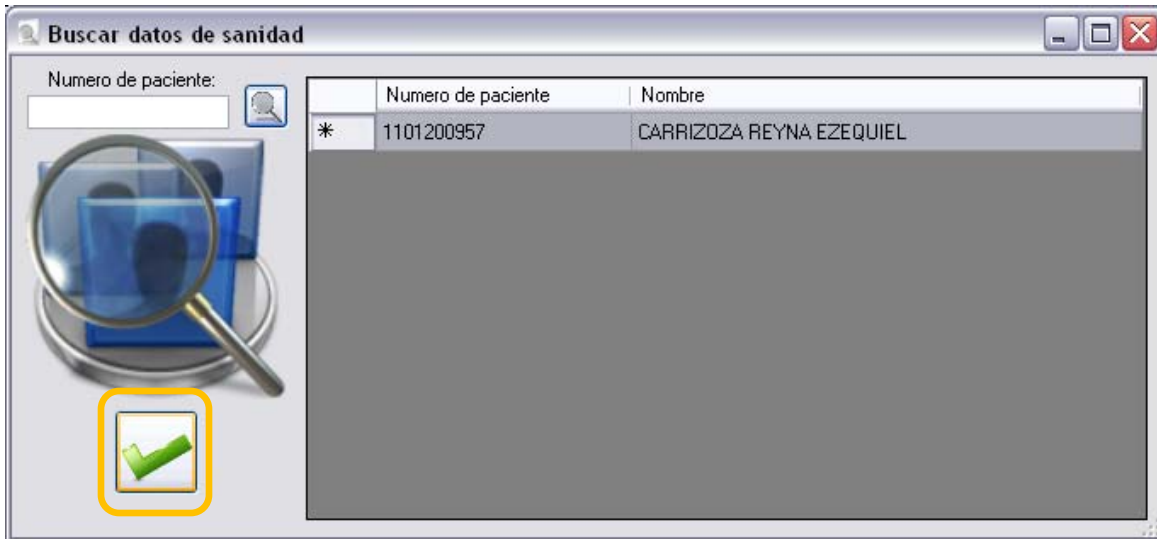
## 12. Credencialización

En el módulo “Credencialización” se puede asignar una tarjeta a un paciente y almacenar sus datos de salud en ella, en este módulo podemos modificar la credencial que se le va a asignar a un paciente. Al hacer clic sobre el botón “Credencialización”, como se muestra a continuación:





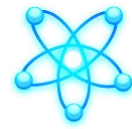
Se mostrara la siguiente pantalla y al dar clic sobre el botón  desplegara todos los pacientes registrados. Se deberá elegir el paciente que se desea asignarle una tarjeta y seleccionar el botón “Aceptar”.



Posteriormente aparecerá toda la información de salud del paciente que se desea asignarle la tarjeta como se muestra a continuación (Colocar la tarjeta sobre el lector ACR120U):







Una vez mostrada esta pantalla validaremos que la información de salud sea la correcta y seleccionar el botón “Aceptar”. Si los datos de salud del paciente se grabaron correctamente en la tarjeta, mostrara el siguiente mensaje:

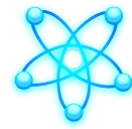


El cual aceptaremos.

### 13. Recolección

En el módulo “Recolección” se puede leer la información la tarjeta, en este módulo podemos recolectar los datos de salud del paciente que se almacenaron en la tarjea. Al hacer clic sobre el botón “Recolección”, como se muestra a continuación:





Se mostrara la siguiente pantalla:

**Recoleccion**

**N° de paciente:**  
**Nombre:**  
**Sexo:**                      **Sangre:**  
**Nivel socioeconomico:**  
**Vivienda:**  
**Grupo Etnico:**  
**Religion:**  
**Enfermedades:**   
**Discapacidades:**   
**Alergias:**   
**Telefono de casa:**                      **Credencial:**

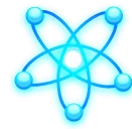
**Aceptar**

Posteriormente colocar el lector cerca de donde se pudes encontrar la tarjeta y seleccionar el botón “Aceptar”. Una vez seleccionado el botón de “Aceptar” se empezara a extraer la información de la tarjeta y será mostrada como se muestra a continuación:

**Recoleccion**

**N° de paciente:** 1101200957  
**Nombre:** EZEQUIEL CARRIZOZA REYNA  
**Sexo:** MASCULINO    **Sangre:** O+  
**Nivel socioeconomico:** C+ CLASE MEDIA ALTA  
**Vivienda:** CASA DE LADRILLO  
**Grupo Etnico:** NINGUNO  
**Religion:** CATÓLICA  
**Enfermedades:** PANCREATTIS  
TRANSTORNOS DEL SUEÑO  
**Discapacidades:** CEGUEA  
DISTONÍA  
**Alergias:** ANTIIBIÓTICOS  
PENICILNA  
**Telefono de casa:** 10-45-09-32                      **Credencial:** 4B 9C BF F7

**Aceptar**



## 14. Cambiar contraseña

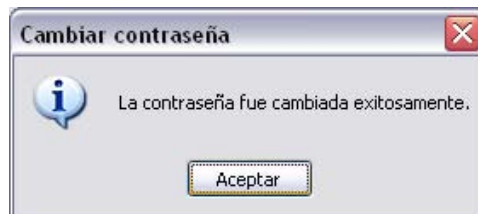
Para modificar la contraseña ingresamos a “Cambiar Contraseña” como se muestra a continuación:

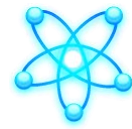


Se mostrará la siguiente pantalla, en la cual capturaremos la contraseña anterior y la nueva. Posteriormente seleccionaremos el botón “Aceptar”:



De haber ingresado las contraseñas correctamente se mostrara el siguiente mensaje que aceptaremos.





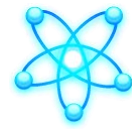
## 15. Acerca de Health Card Manager 2010

Para saber acerca de Health Card Manager 2010, ingresaremos a ver la información de cómo se muestra a continuación:



Se mostrara una pantalla con una pequeña información de HCM 2010, como la que se muestra a continuación:





## 16. Salir de Health Card Manager 2010

Para salir de HCM 2010 ingresar al menú archivo y posteriormente elegir el submenú salir, como se muestra a continuación:



Se mostrara una pantalla en la cual seleccionaremos el botón de “Sí” para salir de HCM 2010.

