

SISTEMA ELECTRONICO DE CONTROL Y TRAZABILIDAD DE MEDICAMENTOS USANDO HARDWARE ARDUINO CON TECNOLOGIA RFID-RC522

JUAN CARLOS OCAMPO GOMEZ
C.C. 1087995276

UNIVERSIDAD TECNOLOGICA DE PEREIRA
FACULTAD DE INGENIERIAS
INGENIERIA ELECTRONICA
PEREIRA 2016

SISTEMA ELECTRONICO DE CONTROL Y TRAZABILIDAD DE MEDICAMENTOS
USANDO HARDWARE ARDUINO CON TECNOLOGIA RFID-RC522

JUAN CARLOS OCAMPO GOMEZ
C.C. 1087995276

DIRECTOR
RODRIGO FRANCO LUNA
INGENIERO ELECTRÓNICO

UNIVERSIDAD TECNOLOGICA DE PEREIRA
FACULTAD DE INGENIERIAS
INGENIERIA ELECTRONICA
PEREIRA 2016

TABLA DE CONTENIDO

LISTA DE TABLAS.....	4
LISTA DE IMAGENES.....	5
RESUMEN.....	6
ABSTRACT.....	7
INTRODUCCION.....	8
PLANTEAMIENTO DEL PROBLEMA.....	9
JUSTIFICACION.....	10
OBJETIVOS.....	11
OBJETIVO GENERAL	11
OBJETIVOS ESPECÍFICOS	11
MARCO TEORICO.....	12
QUE ES EL RFID.....	12
QUE ES ARDUINO	15
QUE ES QT CREATOR	15
QUE ES MARIADB	15
DISEÑO DE LA APLICACIÓN.....	16
Mapa de proceso de la aplicación SCTM-RFID	16
Conexión hardware ADUINO uno con módulo RFID-RC522	17
Aplicar protocolo de comunicación serial entre ARDUINO y programa QT Creator....	19
Creación de un programa para visualización de usuario en programa QT Creator. ...	21
PRESUPUESTO Y FUENTES DE FINANCIACIÓN.....	37
CONCLUSIONES.....	38
BIBLIOGRAFIA.....	39

LISTA DE TABLAS

Tabla 1. Tabla de frecuencias RFID.....	12
Tabla 2. Configuración pines ARDUINO ONU.....	17
Tabla 3. Tabla costo del proyecto.....	37

LISTA DE IMAGENES

Imagen 1. Ejemplo del funcionamiento RFID tag pasivos.	14
Imagen 2. Montaje modulo lector en funcionamiento	18
Imagen 3.Diagrama circuital ARDUINO UNO y módulo RFID-RC522.....	18
Imagen 4.Librerías para controlar puerto serial y módulo RFID.	18
Imagen 5 Objeto de la clase MFRC522.....	18
Imagen 6.Código fuente lectura TAG RFID	19
Imagen 7. Algoritmo configuración puerto serial, librería QSerialPort y QSerialPortInfo	19
Imagen 8.Resultado de datos para la configuración ID vendedor e ID producto.	20
Imagen 9.Archivo .h con las variables contantes inicialización puerto serial	20
Imagen 10. Configuración general para habilitación puerto serial.	20
Imagen 11. Función readSerial().	21
Imagen 12. Menú e entrada SCTM-RFID	22
Imagen 13. Editor de GUI, con sus herramientas.	22
Imagen 14.Código fuente modelos GUI main.ui en C++.	23
Imagen 15.ventana datos del paciente.	24
Imagen 16.validación de registro del paciente.....	24
Imagen 17.Sentencia para ingresar datos a la base de datos en lenguaje MYSQL.	24
Imagen 18.Módulos de ingreso, búsqueda y salida de paciente	25
Imagen 19.Algoritmo principal ventana ingreso paciente.	26
Imagen 20. Sentencia MYSQL, validación serial TAG.....	26
Imagen 21.Ingreso de datos para enviar a la base de datos	27
Imagen 22. Validación de ingreso paciente a la base de datos.....	27
Imagen 23. Consulta MYSQL, condicionad por valor del serial TAG.....	28
Imagen 24. Visualización de la ventana buscar paciente, al momento de realizar la consulta a la base de datos.	28
Imagen 25. Objeto de la clase moviemintosDialog.	29
Imagen 26.Ventanas ingreso, descargue.	29
Imagen 27. Ventana salida paciente.....	30
Imagen 28. Sentencia MYSQL, consulta a base de datos.....	30
Imagen 29. Ventana nota devolución	31
Imagen 30. Ventana Informes	31
Imagen 31.Date edit para rango de fechas.....	32
Imagen 32. Sentencia MYSQL, consulta informe paciente.....	32
Imagen 33. Visualización de la consulta en Tableview.....	32
Imagen 34.Algoritmo creación archivo .CSV	33
Imagen 35. Arreglo de datos para archivo.csv	33
Imagen 36. Protocolo para conexión base de datos HEIDI SQL	34
Imagen 37. Gestor de base de datos HEIDI MYSQL.....	35
Imagen 38. Sentencia MYSQL, creación tabla LOGIN.....	35
Imagen 39. Tabla usuarios LOGIN	35
Imagen 40. Tabla Habitaciones con variables estática.....	36
Imagen 41. Tabla de registros	36

RESUMEN

El proyecto se realizó con la finalidad de llevar el control y trazabilidad de medicamentos de control especial (MCE) o alto costo y la reducción de eventos adversos por errores de medicación con pacientes hospitalizados, esta herramienta permitirá que el personal técnico del Instituto Prestador de Salud (IPS) cuente con un aplicativo fácil y óptimo que permita consultas y ayude a sus labores al momento de la dispensación de medicamentos y minimizar los errores humanos, ya que este procedimiento está definido en forma manual por el personal técnico. Esta aplicación realiza la automatización del proceso de la manipulación de medicamentos MCE o alto costo y brinda a la institución un mejor control de estos medicamentos por fuera de la farmacia.

El proyecto está dividido en tres partes fundamentales: primero, implementación de hardware ARDUINO UNO con módulo RFID-C522 que consta con un sistema de modulación y demodulación para todo tipo de dispositivos pasivos de 13.56Mhz, y una conexión serial hacia la aplicación, segundo, un aplicativo en la plataforma QT Creator firmware C++ que permite la interacción del usuario-servidor y por último una base de datos MARIADB en MYSQL para el registro de datos.

ABSTRACT

The project was carried out with the purpose of bringing the control and traceability of medicinal products subject to special control (MCE) or high cost and the reduction of adverse events by medication errors with hospitalized patients, this tool will allow the technical staff of the Institute Health Provider (IPS) count with an application easy and optimum that allows queries and help your work at the time of the dispensing of medicines and minimize human error, Since this procedure is set manually by the technical personnel. This application performs the automation of the process of handling of medications MCE or high cost and gives the institution a better control of these drugs by outside the pharmacy.

The project is divided into three main parts: first, implementation of ARDUINO hardware one with RFID module-C522 that consists with a system of modulation and demodulation for all types of passive devices of 13.56 MHz, And a serial connection to the application, second, an application on the platform Qt Creator firmware C++ that allows the user interaction-server and finally a database MARIADB in MySQL for the registration of data.

INTRODUCCION

En el año 2015 un estudio llamado "Mejorar La Seguridad En La Utilización De Medicamentos" del Ministerio de Salud y Protección Social de Colombia, indica que los errores de medicación constituyen un 19,4 % del total de lesiones en los hospitales, siendo una de las principales causas de eventos adversos. Este estudio muestra las falencias al momento de suministrar medicamentos a los pacientes lo cual se debe a la falta de mecanismos de trazabilidad, control y monitoreo que sean eficientes y reduzcan considerablemente estos riesgos. En la actualidad el personal hospitalario realiza el control, suministro y validación de medicamentos dando posibilidad a cometer un error, donde se vuelve más crítico en medicamentos de control especial (MCE), los cuales tienen potencial de causar abuso o dependencia. Estos son restringidos ya que también pueden crear un gran riesgo para la salud si se usan inadecuadamente. Las lesiones por mala utilización de medicamentos suceden a que naturalmente en todo el mundo se presentan malas prácticas debido a diferentes factores físicos o psicológicos del personal asistencial. Obligando a los gobiernos y entidades relacionadas, establecer protocolos que disminuyan los eventos adversos.

Los avances tecnológicos en sistemas de identificación automática RFID crean nuevas formas de control y han permitido una mejor trazabilidad no sólo en el área de la medicina sino también en cualquier tipo de servicio. Partiendo de los sistemas ya conocidos como la codificación digital (código de barras), se genera una nueva tecnología inalámbrica llamada RFID (Radio Frequency Identification) o identificación por radiofrecuencia, el cual es un sistema de almacenamiento y recuperación de datos remoto que usan dispositivos denominados etiquetas, tarjetas, o tags para cumplir su propósito de identificar algún objeto sin necesidad de contacto. Dicha etiqueta brinda la posibilidad de lectura y escritura adaptándose a necesidades de esta problemática. La tecnología RFID se puede aplicar al monitoreo de los medicamento para llevar un control más estricto y disminuir en gran medida errores humanos a eventos adversos. Se implementará un software acompañado de una base de datos, enlazado con el hardware ARDUINO con receptor RFID-RC522, realizando prueba piloto en el Hospital Universitario San Jorge de la ciudad de Pereira.

PLANTEAMIENTO DEL PROBLEMA

Una problemática importante para las Instituciones Prestadoras de Salud (IPS) de nuestro país son los errores en la administración de medicamentos, uno de ellos y más frecuente son los accidentes por confusión de paciente ("paciente equivocado"); el error se da cuando el personal de salud procede a la distribución y administración de medicamentos a cada uno de los pacientes, lo que conlleva un alto riesgo para la salud de los mismos, y puede no tener una atención oportuna para tratar su patología donde esta se agudiza mucho más si no es detectado a tiempo, finalmente se debe tratar el medicamento que le fue suministrado erróneamente para evitar posibles eventos adversos, ya que sería posible que no fuera solo un paciente afectado sino dos, llevándolo al peor de los casos.

En la actualidad el Hospital General, North York en Toronto Canadá, es el primer hospital en Canadá desde septiembre del 2014 en implementar la tecnología RFID con identificación de medicamentos, con un dispositivo llamado "The Intelliguard® Kit and Tray Management System from MEPS Real-Time", es un sistema gestor de bandejas, que permite etiquetar cada uno de los medicamentos con etiquetas RFID y ubicarlos en bandejas que posteriormente, se introducen en un gabinete el cual cuenta con antenas receptoras RFID y un sistema de gestor de datos por medio de un computador para identificar cada uno de los medicamentos que contiene la bandeja. Un estudio de calidad realizado por el hospital fue capaz de lograr el 100 por ciento de exactitud, mostrando la eficiencia de esta tecnología que reduce los errores de medicación y mejora la seguridad del paciente.

Mediante este proyecto se pretende diseñar e implementar un prototipo electrónico con hardware ARDUINO y periférico RFID-RC522, que permita llevar trazabilidad y control mediante una interfaz de usuario desarrollado en la plataforma QT Creator permitiendo fácil interacción con el personal hospitalario. Esta información se ingresa a un gestor de base de datos MYSQL que posteriormente permite mostrar datos representativos de movimiento de cada medicamento y su aplicación al paciente.

Gracias a la plataforma QT Creator y ARDUINO son software libre, se puede concebir un dispositivo económico, escalable y versátil con bajos costos de implementación, este proyecto garantizara el mejoramiento de los eventos adversos informados anteriormente.

JUSTIFICACION

El proyecto está destinado a la falta de sistematización que entidades hospitalarias de salud, no cuentan con un sistema de trazabilidad y control de medicamentos MCE sistematizado, al momento de ser dispensados por farmacia, este control lo realiza de forma manual y proclive al error humano, también eventos adversos al momento de la aplicación del medicamento al paciente, en la actualidad se encuentra una compañía canadiense MEPS REAL-TIME, INC que ha realizado el primer sistema de control y trazabilidad de inventarios realizando una implementación de etiquetas RFID que permite la identificación de medicamentos y realización de inventarios en forma exacta, esto lo consigue realizando el etiquetado de cada uno de los medicamentos con etiquetas muy pequeñas que permiten su fácil adaptación a cualquier medicamento, luego de tener cada medicamentos etiquetado, se debe colocar este en un kit de bandejas y posteriormente ingresan en un dispositivo con receptores RFID que lee cada etiqueta por separado, permitiendo su identificación al instante, este dispositivo cuenta con una interfaz gráfica que muestra el inventario al instante. Este kit fue lanzado en el Hospital General de North York en Canadá en el año 2015, dando como resultados contundentes 100% de efectividad en su funcionamiento. Pero tiene una desventaja es su alto costo para su implementación.

En los estudios realizados por el Ministerio de Salud y Protección Social de Colombia, se observa la necesidad de tener un sistema que permita ayudar al personal de la salud en el control y trazabilidad de medicamentos por la IPS, también se iniciaría el proyectos para que la tecnología RFID que en muchos lugares del mundo como es el caso del hospital de Canadá es un caso exitoso.

Este proyecto quiere ir un poco más allá y no solo garantizar la trazabilidad y control de medicamentos sino que sea una herramienta útil para el personal hospitalario y evitar que el paciente sufra eventos adversos que afecten su salud. Generar una eficiencia del 100% y reducción de eventos adversos. Gracias a que se desarrolla en plataformas como hardware ARDUINO UNO con módulo RDIF-RC522, Qt Creator y MARIADB programas de software libres, software libre es la denominación del software que respeta la libertad de todos los usuarios que adquirieron el producto y, por tanto, una vez obtenido el mismo puede ser usado, copiado, estudiado, modificado, y redistribuido libremente, sin ningún costo, esto disminuye considerablemente la implementación del SCTM-RFID con respecto a otras compañías que solo brinda el soporte de inventarios como MEPS REAL-TIME, INC. Por las cuales tiene una gran inversión por acceder a sus productos.

OBJETIVOS

OBJETIVO GENERAL

- Implementar un sistema electrónico utilizando hardware ARDUINO con lector RFID-RC522 e interfaz de programación QT Creator (Community), para el control y trazabilidad de 10 medicamentos.

OBJETIVOS ESPECÍFICOS

- Implementación de hardware ARDUINO con lector de RFID-RC522, como lector de TAG para caja de medicamentos.
- Aplicar protocolo de comunicación serial entre ARDUINO y programa QT Creator.
- Creación de un programa para visualización de usuario en programa QT Creator.
- Aplicación de protocolo de comunicación de entre los programas QT Creator y gestor de base de datos MARIADB (HeidiSQL).
- Generar una base de datos MYSQL MARIADB (HeidiSQL) para ingreso de información de cada registro siniestrado por el TAG RFID.
- Verificar la eficiencia del sistema mediante test de funcionamiento de la tecnología en el Hospital Universitario San Jorge de la ciudad de Pereira.

MARCO TEORICO

QUE ES EL RFID

RFID significa (Radio Frequency Identification), en español Identificación por Radio Frecuencia, "RFID" hace referencia a un tipo de tecnología de intercambio inalámbrico de datos. La lectura y grabación de los datos se realiza a partir de un chip conectado a una antena que recibe señales de radiofrecuencia desde una antena receptora o emisora que permite de lectura y grabación. El intercambio de datos se produce automáticamente, sin que ningún operador tenga que intervenir para activar la lectura de RFID.

Esta tecnología opera en tres tipos de frecuencias: baja frecuencia (LF), alta frecuencia (HF), banda (UHF).

	Baja Frecuencia(LF Low Frequency)	Alta Frecuencia (HF High Frequency)	Bandas(UHF Ultra High Frequency)
Rango de frecuencias	30-300 KHz	3-300MHz	300MHz-3GHz
Rango de operatividad dispositivos	125KHz	13,5MHz	900-915MHz
Distancia cobertura	0,01 metros	0,01-0,10 metros	Hasta 12 metros

Tabla 1. Tabla de frecuencias RFID.

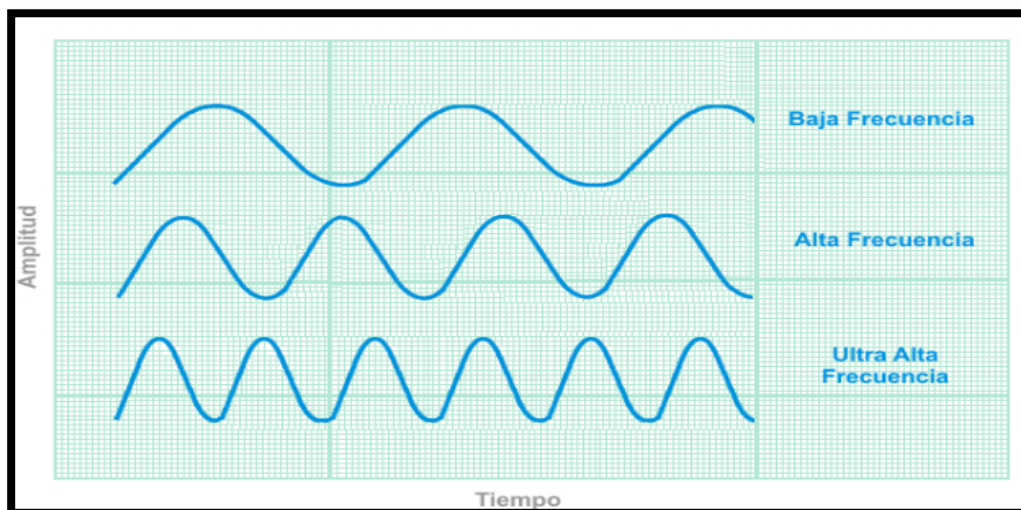


Ilustración 1. Frecuencias RFID.

TIPOS DE RFID

En el mercado existen tres tipos de RFID

- **Activos:** Estos contienen fuentes propias y capaces de alcanzar kilómetros, pero su alto costo hace difícil su acceso en la actualidad.
- **Pasivas:** No contienen fuente propia, el alcance de lectura es proporcional a la antena receptora, son las más usadas del mercado por su precio.
- **Semi-pasivas:** es una combinación de las dos, brindan un rango de cobertura amplio

HISTORIA RFID

La tecnología RFID tiene más de 50 años de existir, se implementó en la segunda guerra mundial por la armada británica, donde se usaban ondas de radio frecuencia para identificar sus aviones en el campo de batalla de los aviones enemigos. Luego de la guerra no fue usada por sus altos costo para su implementación, en la última década gracias a los avances en tecnológicos y nuevos requerimientos de la industria, la tecnología por radio frecuencia RFID ha reducido sus costos de fabricación y es posible acceder a esta tecnología con mayor facilidad, aunque no es una tecnología nueva en los últimos años hay un auge demostrando su adaptabilidad a varios campos de la industria y su fácil implementación.

COMO FUNCIONA

Los sistemas RFID constan de etiquetas o tags adhesivas o en forma más duradera como tarjetas o manillas. El lector envía una señal de radio que es recibida por todos los tags que se encuentra en el rango del campo de radiofrecuencia sintonizado con dicha frecuencia. Los tags reciben la señal a través de sus antenas y responden transmitiendo los datos que tienen almacenado. El tag puede almacenar muchos tipos de datos, según la memoria de almacenamiento que contenga el tag desde bits hasta Mbyte. El dispositivo de lectura/ escritura recibe la señal del tag a través de su antena, la descodifica y transfiere los datos al sistema informático a través de una conexión de cable o inalámbrica.

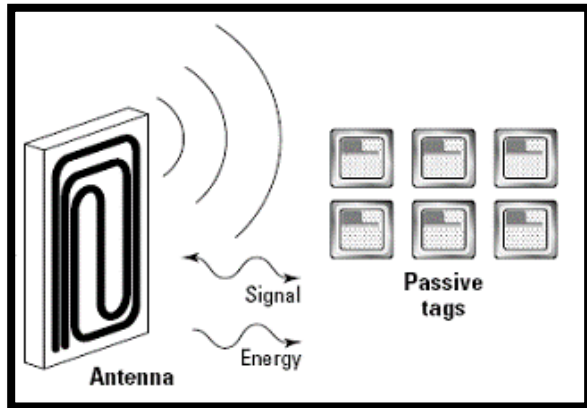


Imagen 1. Ejemplo del funcionamiento RFID tag pasivos.

VENTAJAS RFID

La tecnología RFID posee una serie de características que la convierten en la mejor alternativa frente a otros tipos de sistemas de identificación como son:

- La RFID permite controlar y capturar datos en entornos inadecuados que es imposible ver la etiqueta, ya que no es necesario tener la etiqueta a la vista para ser leída.
- Esta tecnología permite realizar más de mil lecturas por segundo, ofreciendo una alta velocidad y una gran precisión.
- Puede contener mayor información en Bytes que un código de barra.
- Rapidez en la comunicación de datos.
- Los datos de un tag RFID (también conocido como tag) puede ser sobre-escribir
- La tecnología RFID no necesita una línea directa de visión entre la etiqueta y el lector, lo que la hace adecuada para muchas aplicaciones en las que no se pueden utilizar códigos de barras
- Miles de empresas de numerosos sectores productivos han explotado las ventajas de la identificación por radiofrecuencia para desarrollar operaciones que controlan procesos, facilitan datos precisos en tiempo real, realizan el seguimiento de bienes e inventarios y reducen los requisitos de mano de obra.

QUE ES ARDUINO

ARDUINO es una compañía de software y hardware libre, el cual desarrolla placas integrando un microcontrolador, entradas/salidas análogas y digitales, con conexión SPI, en un entorno de desarrollo (IDE) abierto que permite realizar prototipos para inicios de algún proyecto escalable, con múltiples aplicaciones y entorno amigables para cualquier persona.

QUE ES QT CREATOR

QT Creator es un entorno de desarrollo integrado (IDE) multiplataforma en lenguaje C++, que permite al usuario desarrollar multiplex aplicaciones según sean las necesidades con un entorno gráfico QDesing, software libre con licencia GLP gratuita.

QUE ES MARIADB

MARIADB es un sistema de bases de datos derivado de MySQL con licencia GLP, con gestor de base de datos HEIDI SQL software libre para implementación gratuita, conexión TCP/IP o SSH.

DISEÑO DE LA APLICACIÓN

Mapa de proceso de la aplicación SCTM-RFID

Este mapa de procesos resumen los cinco ítem principales para la creación del programa Sistema de Control y Trazabilidad de Medicamentos RFID (SCTM-RFID), se explicara cada uno de los ítem y las forma de implementación de cada uno de los procesos necesarios para logra su culminación.

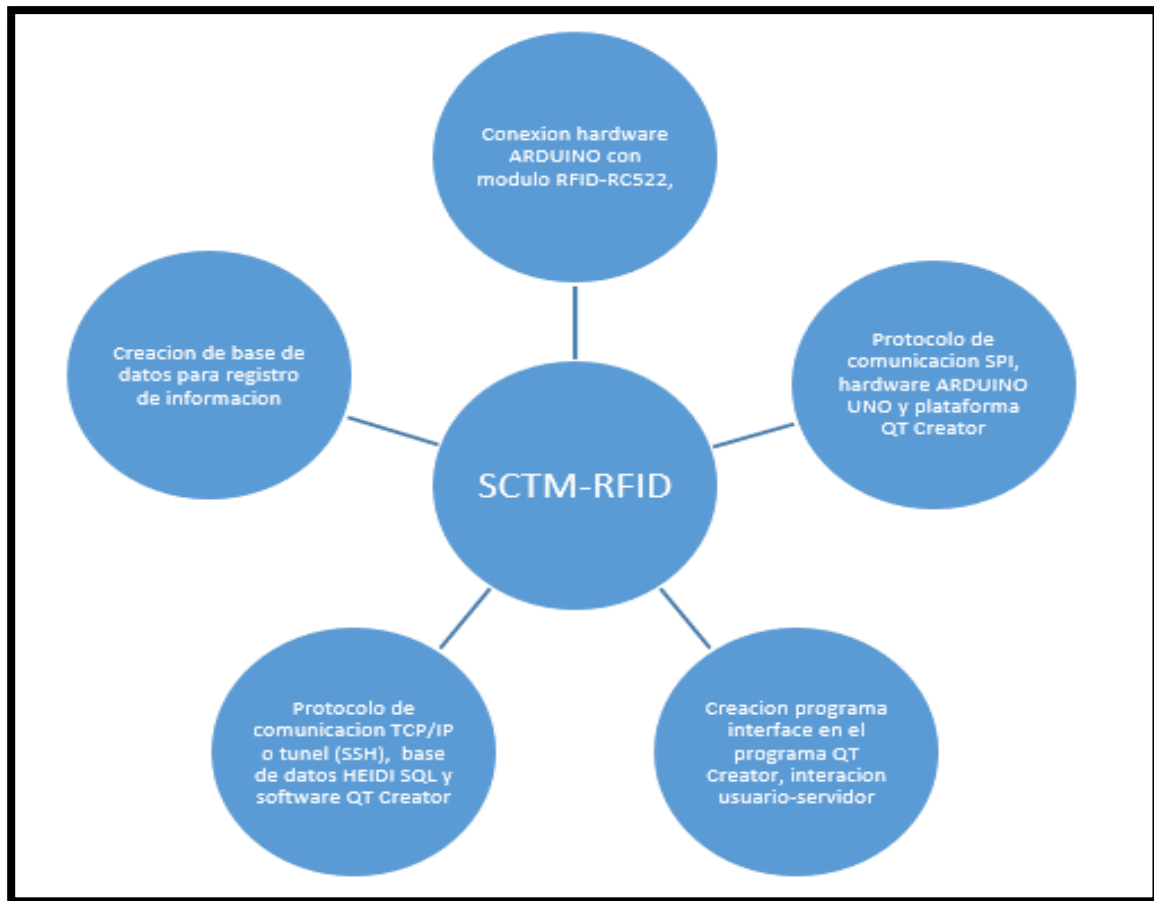


Ilustración 2. Mapa de proceso para la generación Sistema de Control y Trazabilidad de Medicamentos RFID.

Se explicara el diseño del proyecto basados en los 5 ítems anteriores mostrando el cumplimiento de ellos.

Conexión hardware ARDUINO uno con módulo RFID-RC522

Para este proyecto se decide usar el hardware ARDUINO UNO con modulo LECTOR RFID-RD522 por su conexión SPI.

ARDUINO es una plataforma electrónica, con código abierto (open-source) basada en hardware y software flexibles y fáciles de usar, a continuación mostraremos como se realizaron las conexiones entre el ARDUINO UNO y el modulo lector RFID-RC522.

ARDUINO UNO	RFID RC522
DIGITAL PIN 9	RST
DIGITAL PIN 5	LED STANBY
DIGITAL PIN 6	LED LECTURA
DIGITAL PIN 12	MISO
DIGITAL PIN 11	MOSI
DIGITAL PIN 13	SCK
GND	GND
3.3 V	3.3V
N/A	IRQ

Tabla 2. Configuración pines ARDUINO UNO

MOSI (Master Out Slave In): Línea utilizada para llevar los bits que provienen del maestro hacia el esclavo.

MISO (Master In Slave Out): Línea utilizada para llevar los bits que provienen del esclavo hacia el maestro.

SCK (Clock): Línea proveniente del maestro encarga de enviar la señal de reloj para sincronizar los dispositivos e inicial al dispositivo esclavo.

RST: Reset

Gracias a su protocolo de conexión SPI serial que tiene el modulo nos permite interactuar y poder realizar la lectura que en los TAG de forma rápida y correcta. Para validar su estado se instalaron dos LED informativos que permite verificar el estado de la lectura del TAG. El color rojo informa que se encuentra en un estado STANDBY o encendido y el LED de color verde nos informa la lectura correcta del código de la tarjeta o TAG correctamente. Ver (Imagen 3.Diagrama circuital ARDUINO UNO y módulo RFID-RC522.



Imagen 2. Montaje modulo lector en funcionamiento

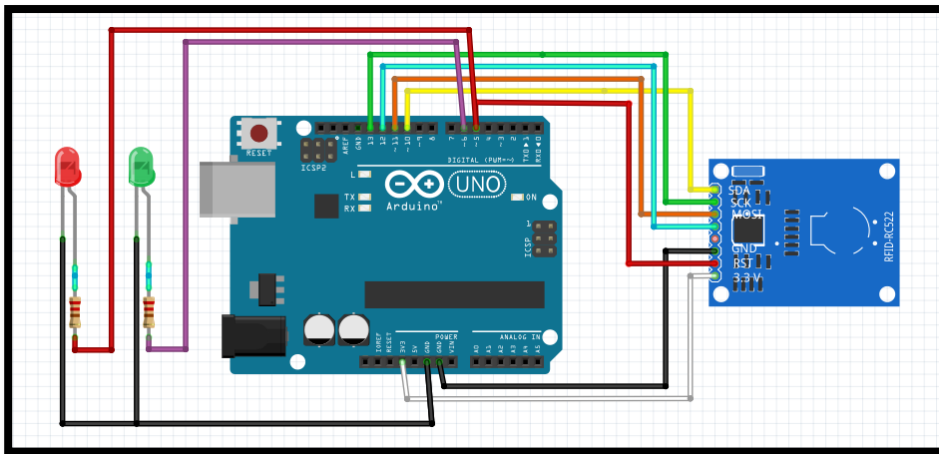


Imagen 3. Diagrama circuital ARDUINO UNO y módulo RFID-RC522.

Para su ejecución ingresan las librerías del puerto serial y del módulo RFID-RC522.

```
#include <SPI.h>
#include <MFRC522.h>
```

Imagen 4. Librerías para controlar puerto serial y módulo RFID.

Realizamos el objeto del módulo RFID-RC522 gracias a la librería MFRC522.

```
//-----objeto del modulo RFID-RC522-----
MFRC522 mrc522(SS_PIN, RST_PIN); //Creamos el objeto para el RC522
```

Imagen 5 Objeto de la clase MFRC522

Este código muestra su practicidad y funcionalidad que nos permite la plataforma ARDUINO, se realiza una validación con el fin de verificar el momento que se encuentre una tarjeta o TAG cerca del lector y enviar por puerto serial el número de serie.

```
if ( mrc522.PICC_IsNewCardPresent() )
{
    //Seleccionamos una tarjeta
    if ( mrc522.PICC_ReadCardSerial() )
    {
        digitalWrite(led_Stanby,false);
        digitalWrite(led_Lectura,true);
        Serial.print(",");
        for (byte i = 0; i < mrc522.uid.size; i++)
        {
            Serial.print(mrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
            Serial.print(mrc522.uid.uidByte[i], HEX);
        }
    }
}
```

Imagen 6. Código fuente lectura TAG RFID

Aplicar protocolo de comunicación serial entre ARDUINO y programa QT Creator.

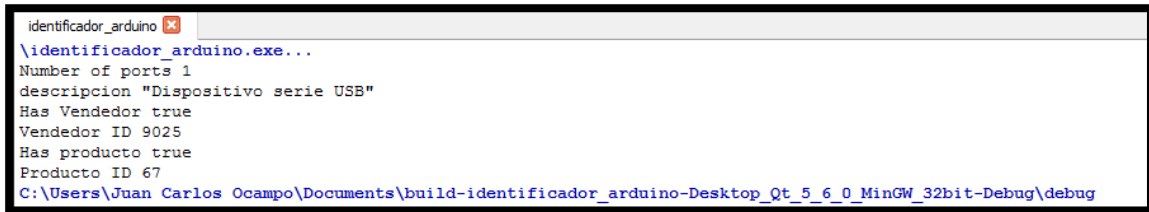
QT Creator contiene librerías de comunicación serial llamada QSerialport esta nos permite la comunicación con otros dispositivo externo que contenga este mismo protocolo de comunicación, que es compatible con ARDUINO UNO, esta librería permite la sincronización con dispositivos y la transferencia de datos sea en tiempo real, para realizar este parámetro es necesario conocer el dispositivo a conectar para configurar los parámetros que nos exigen. Se mostrara como realizar la comunicación necesaria para interactuar entre el ARDUINO UNO y el programa QT Creator.

Para esta conexión es necesario conocer el ID del vendedor y del producto, parámetros necesario para la librería QSerialPort. Se muestra a continuación el código de validación de ID para el producto, se trata de validar el vendedor y el productor que se va a conectar para realizar la conexión posterior

```
11 ui->setupUi(this);
12 qDebug()<<"Number of ports"<<QSerialPortInfo::availablePorts().length();
13 foreach(const QSerialPortInfo &serialPortInfo, QSerialPortInfo::availablePorts())
14 {
15     qDebug()<<"descripcion"<<serialPortInfo.description();
16     qDebug()<<"Has Vendedor"<<serialPortInfo.hasVendorIdentifier();
17     qDebug()<<"Vendedor ID"<<serialPortInfo.vendorIdentifier();
18     qDebug()<<"Has producto"<<serialPortInfo.hasProductIdentifier();
19     qDebug()<<"Producto ID"<<serialPortInfo.productIdentifier();
20 }
21
```

Imagen 7. Algoritmo configuración puerto serial, librería QSerialPort y QSerialPortInfo

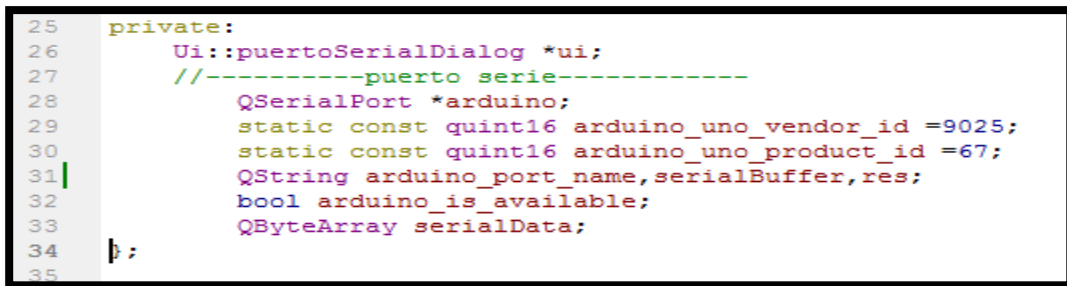
Podemos validar que el dispositivo si es aceptado por la plataforma QT Creator y continuamos a realizar la conexión definitiva para seguir con el programa.



```
identificador_arduino x
\identificador_arduino.exe...
Number of ports 1
descripcion "Dispositivo serie USB"
Has Vendedor true
Vendedor ID 9025
Has producto true
Producto ID 67
C:\Users\Juan Carlos Ocampo\Documents\build-identificador_arduino-Desktop_Qt_5_6_0_MinGW_32bit-Debug\debug
```

Imagen 8. Resultado de datos para la configuración ID vendedor e ID producto.

Ya con la información del vendedor y producto se colocan como variables contantes en nuestra cabecera .h, realizamos el objeto *arduino con todos los atributos de la clase **QSerialport**.



```
25 private:
26     Ui::puertoSerialDialog *ui;
27     //-----puerto serie-----
28     QSerialPort *arduino;
29     static const quint16 arduino_uno_vendedor_id =9025;
30     static const quint16 arduino_uno_producto_id =67;
31     QString arduino_port_name, serialBuffer, res;
32     bool arduino_is_available;
33     QByteArray serialData;
34 ;
```

Imagen 9. Archivo .h con las variables contantes inicialización puerto serial

Observamos la variables necesarias para habilitar el puerto serial en QT Creator, la librería como antes mencionada, que realiza este procedimiento se llama QSerialport. Ahora lo tenemos en el objeto *arduino, inicializamos el puerto con los datos ya preestablecidos posteriormente y abrimos el puerto serial para realizar su lectura.



```
12 ui->setUpUi(this);
13 //-----SerialPort-----
14 qDebug()<<"entro buscar";
15 arduino_is_available=false;
16 arduino_port_name="";
17 arduino =new QSerialPort;
18 serialBuffer="";
19 foreach(const QSerialPortInfo &serialPortInfo, QSerialPortInfo::availablePorts())
20 if(serialPortInfo.hasVendorIdentifier() && serialPortInfo.hasProductIdentifier())
21 {
22     if(serialPortInfo.vendorIdentifier() == arduino_uno_vendedor_id)
23     {
24         if(serialPortInfo.productIdentifier() == arduino_uno_producto_id)
25         {
26             arduino_port_name = serialPortInfo.portName();
27             arduino_is_available = true;
28         }
29     }
30 }
31
32 if(arduino_is_available)
33 {
34     arduino->setPortName(arduino_port_name);
35     arduino->open(QSerialPort::ReadOnly);
36     arduino->setBaudRate(QSerialPort::Baud9600);
37     arduino->setDataBits(QSerialPort::Data8);
38     arduino->setParity(QSerialPort::NoParity);
39     arduino->setStopBits(QSerialPort::OneStop);
40     arduino->setFlowControl((QSerialPort::NoFlowControl));
41     connect(arduino, SIGNAL(readyRead()),this,SLOT(readSerial()));
42 }
43 }
44 else
45 {
46     QMessageBox::warning(this,"Error puerto","verificar puerto del arduino");
47 }
```

Imagen 10. Configuración general para habilitación puerto serial.

```

60 void puertoSerialDialog::readSerial()
61 {
62     QStringList bufferSplit =serialBuffer.split(",");
63     if (bufferSplit.length()<3)
64     {
65         serialData = arduino->readLine();
66         serialBuffer += QString::fromStdString((serialData.toStdString() ));
67         qDebug()<<"buffer";
68     }
69     else
70     {
71         ui->lineEdit->setText (res.mid(1,11));
72         serialBuffer="";
73         res=bufferSplit[1];
74         emit ps(res);
75     }

```

Imagen 11. Función readSerial().

Con la función **connect** nos permite conectar una señal con una función, esto es muy útil ya que al momento de generar una lectura el puerto (SIGNAL readyRead()) podemos conectar a una función (readSerial()), que nos permite manipular dicha variable.

Ya que la plataforma QT Creator es basada en programación orientada objetos, realizamos una función **readSerial()**, con esta verificar lo encontrado en el puerto serial, esta función realiza una división del arreglo que se encuentra en el puerto, para leer el dato del serial de la tarjeta RFID o TAG, al momento de ser leído por el modulo se realiza la lectura y se añade una coma (","), con el fin de dividir el arreglo e identificar el número que deseamos (serial del TAG) y guardarlo en una variable específica.

Creación de un programa para visualización de usuario en programa QT Creator.

La interfaz gráfica se realiza en el programa QT Creator es un IDE que nos brinda la facilidad de realizar por completo nuestro código desde cero con la gran ventaja que es multiplataforma en base C++. La interfaz que se diseño es basado completamente a ayudar al usuario, que lo vean como una herramienta que le ayudara en su trabajo y no algo molesto, que le quite más tiempo a sus labores, siguiendo con esta idea, se mostrara el paso a paso a llegar al programa SCTM-RFID.

La aplicación para realizar una mejor explicación se desarrolló en 4 fases, ventanas de acceso para cada solicitud:

La primera ventana para realizar la autenticación del usuario, ver la siguiente imagen.

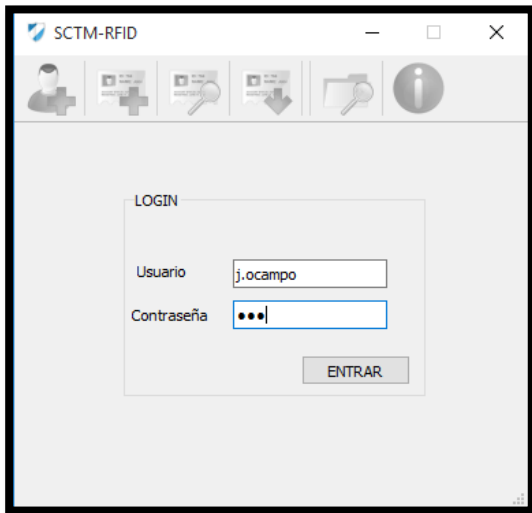


Imagen 12. Menú e entrada SCTM-RFID

Esto se logra ya que la plataforma QT Creator contiene una herramienta demasiado útil para los programadores llamada Qt Designer, es la herramienta que sirve para el diseño y la construcción de interfaces gráficas de usuario (GUI) con Qt widgets. Esta realiza cuadros o ventanas personalizadas a estilos y necesidades que el proyecto lo necesite.

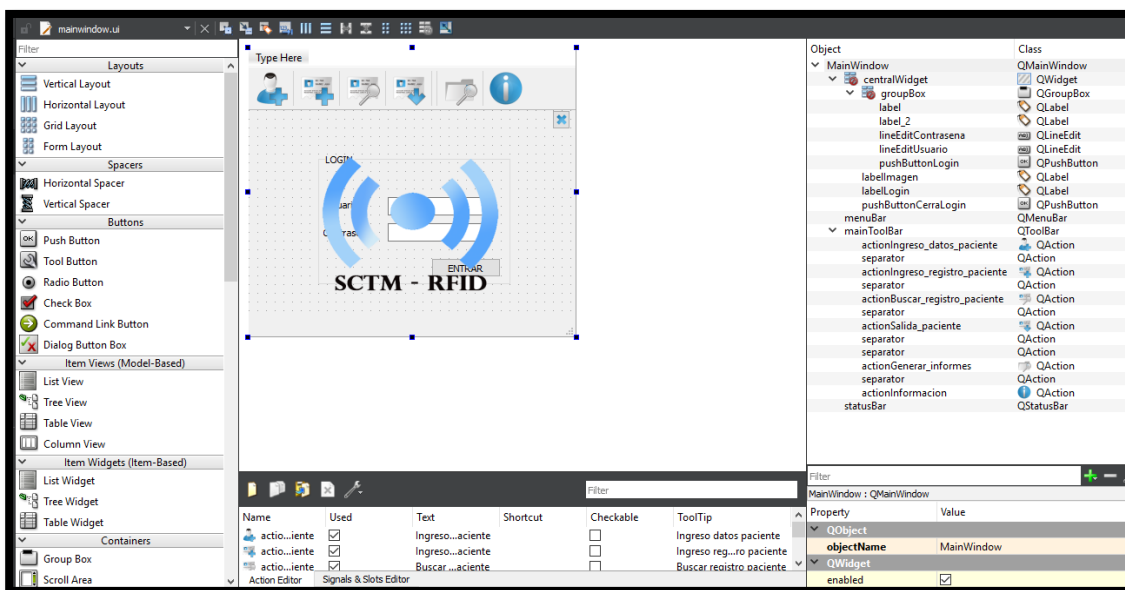


Imagen 13. Editor de GUI, con sus herramientas.

Este GUI (interfaz gráfica) crea un código C++ que nos admite si es necesarios cambiar cualquier atributo cómo el estilo, forma, diámetro del objeto, color, fuente y según sea su objeto, desde el lenguaje C++ como se muestra en la siguiente figura.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ui version="4.0">
3 <class>MainWindow</class>
4 <widget class="QMainWindow" name="MainWindow">
5 <property name="geometry">
6 <rect>
7 <x>0</x>
8 <y>0</y>
9 <width>376</width>
10 <height>331</height>
11 </rect>
12 </property>
13 <property name="minimumSize">
14 <size>
15 <width>376</width>
16 <height>331</height>
17 </size>
18 </property>
19 <property name="maximumSize">
20 <size>
21 <width>376</width>
22 <height>331</height>
23 </size>
24 </property>
25 <property name="windowTitle">
26 <string>SCTM-RFID</string>
27 </property>
28 <property name="windowIcon">
29 <iconset resource="imagenes.qrc">
30 <normaloff>:/shield-icon.png</normaloff>:/shield-icon.png</iconset>
31 </property>
32 <widget class="QWidget" name="centralWidget">
33 <widget class="QGroupBox" name="groupBox">
34 <property name="geometry">
35 <rect>
36 <x>80</x>
37 <y>50</y>
```

Imagen 14. Código fuente modelos GUI main.ui en C++.

Cada una de las ventanas se realiza con la herramienta Qt Designer, ya que esta nos muestra exactamente lo que deseamos, y es mucho más óptimo al momento de no realizar manualmente la interfaz.

Se realizó las ventanas o interfaces como antes mencionados, de modo que el programa sea lo más óptimo y simple sin perder su funcionalidad, por el grado de ocupación de los usuarios que está destinado este proyecto.

Como se muestra a continuación la explicación de cada una de la ventanas con algunos de restricciones que se necesitaba para logra el objetivo

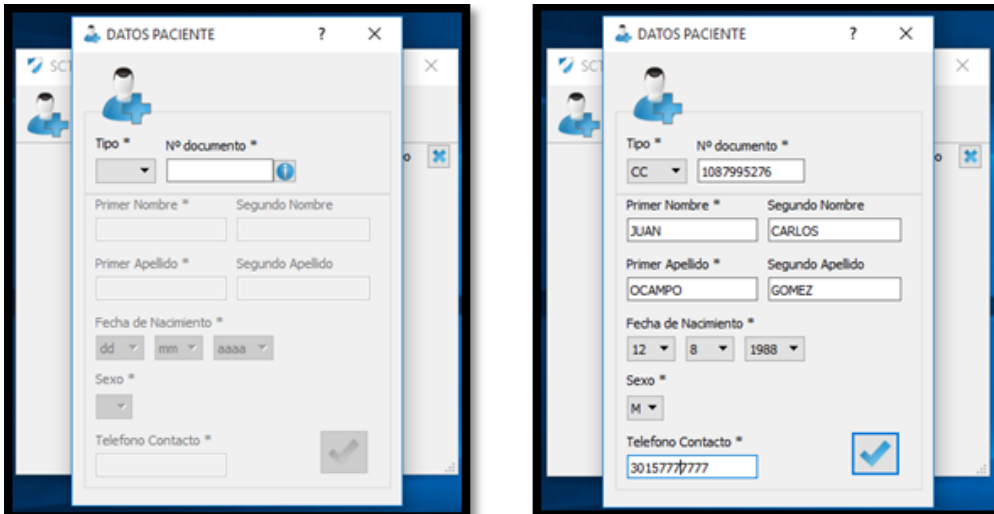


Imagen 15.ventana datos del paciente.

Como se ve en las siguientes dos imágenes se muestra el ingreso de los datos del paciente, con este se realiza en Qt Designer, llevando una estética simple y funcional, esta ventana contiene requerimientos como los datos básicos como: Tipo de documento, nombre, apellidos, fecha de nacimientos para posteriormente realizar la operación de la edad y por ultimo un número de contacto. Se resalta que antes de llenar esta información se debe validar el documento para no realizar dobles ingresos o pérdida de tiempo por parte del personal ingresando un paciente que ya se encontraba registrado en la base de datos.

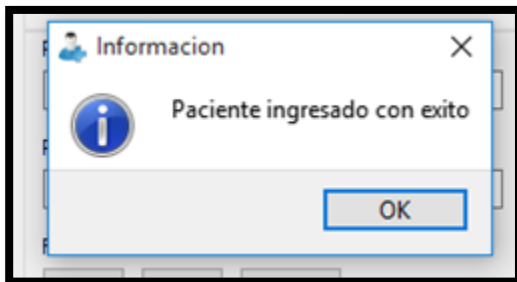


Imagen 16.validación de registro del paciente.

Como es de la programación direccionada a objetos para esta parte se realiza en el encabezado las variables necesarias para ser enviadas a la base de datos HEIDI MYSQL, como se indica en el siguiente algoritmo, cada una de estos ingreso se realiza por medios de funciones privadas y solo acceder a ella por el siguiente método.

```

QSqlQuery qry;
qry.prepare("INSERT INTO paciente (cedula,nombre,nombre2,apellido,apellido2,fecha_nacimiento,sexo,telefono,estado) "
           "values ('"+cedula+"','"+nombre+"','"+nombre2+"','"+apellido+"','"+apellido2+"','"+fechaNacimiento+"',"
           "''"+sexo+"','"+telefono+"','in')");

```

Imagen 17.Sentencia para ingresar datos a la base de datos en lenguaje MYSQL.

Se realiza un barrido por cada QlineEdit (cedula, nombres, apellido, teléfono, etc.), barras de texto para realizar una verificación que se encuentren cada uno de los datos solicitados, si uno de estos no se encuentra realiza un return al instante y envía un mensaje informativo para que el usuario pueda realizar la corrección y seguir con el ingreso.

Cada una de las ventanas consta de 3 módulos de mayor importancia que es el ingreso del paciente al sistema, búsqueda (ingreso, egreso), salida del paciente. Estos se diseñaron para que contengan la siguiente información:

TAG asignado

Datos del paciente

Habitación

Diagnostico en I10x

Medicamentos a cargar

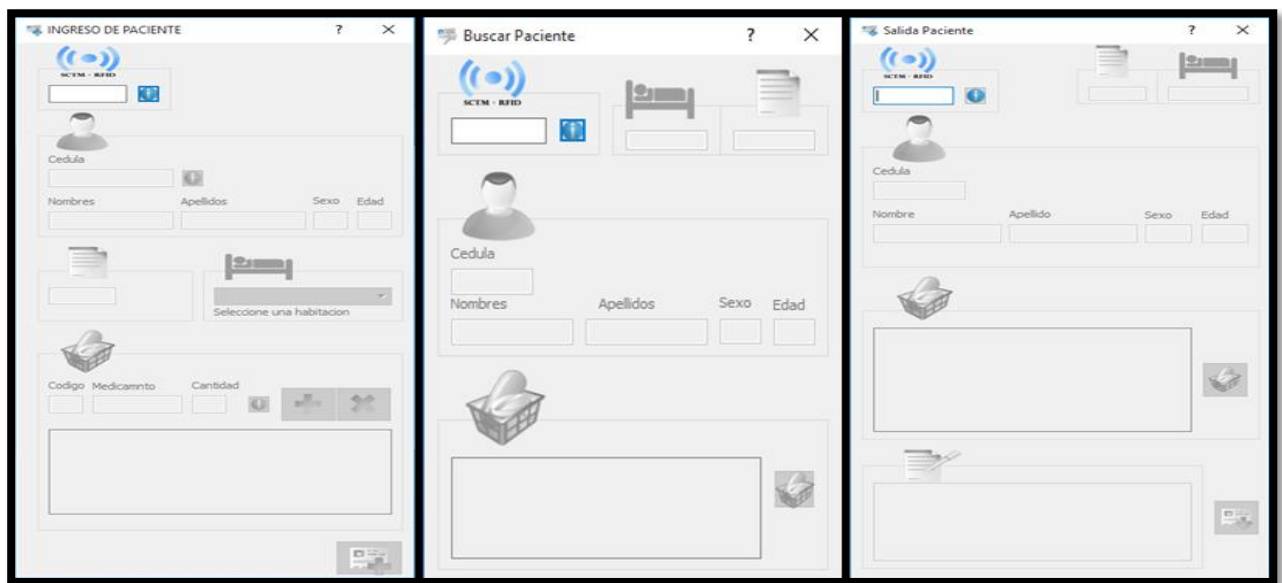


Imagen 18. Módulos de ingreso, búsqueda y salida de paciente

Como se puede ver las tres ventanas son muy similares con el fin de realizar una armonía y generar un ambiente intuitivo para el usuario.

Para realizar estas ventanas mostraremos su parte fundamental para cada una de ellas y su código para realizar.

VENTANA INGRESO PACIENTE

Esta ventana contiene una serie de restricciones, ya que sin ellas se puede proceder en conflictos con la base de datos y generar datos redundantes. Se muestra en el siguiente algoritmo en el archivo.cpp, se verifica una consulta a la base de datos en lenguaje MySQL, que el tag ingresado se encuentre habilitado para su uso.

```
//-----consulta id_paciente, trae los valores de la tabla paciente
QString cedula,estado;
cedula=ui->lineEditCedula->text();
QSqlQuery qry;
qry.prepare("SELECT *,DATEDIFF(curdate(),fecha_nacimiento)/365 "
           "FROM paciente "
           "WHERE cedula='"+cedula+"'");
if(qry.exec())
{
    ui->groupBoxHabitacion->setDisabled(true);
    ui->groupBoxMedicamento->setDisabled(true);
    ui->lineEditNombre->clear();
    ui->lineEditApellido->clear();
    ui->lineEditEdad->clear();
}
```

Imagen 19. Algoritmo principal ventana ingreso paciente.

Si estos datos son correctos, lo que se realiza por medio del archivo .ui es mostrar datos al usuario, también estos datos se guardan en variables específicas para realizar como registro a la base de datos.

Para realizar el ingreso a la base de datos de los medicamentos, este se valida por medio de una consulta realizando un objeto qry de la clase QSqlQuery, con esto podemos acceder a un arreglo a datos ya establecido en tablas de las mismas, en la base de datos que retornan como un arreglo de datos.

```
QSqlQuery qry;
qry.prepare("SELECT * FROM medicamentos WHERE codigo='"+codigo+"'");
if(qry.exec())
{
    while(qry.next())
    {
        ui->lineEditNomMed1->setText(qry.value(2).toString());
        id_medicamento=qry.value(0).toString();
    }
}
```

Imagen 20. Sentencia MYSQL, validación serial TAG

Al momento de acceder a estos datos podemos pasar nuestro arreglo a variables QString para realizar la validación, si estos datos se encuentran duplicados por el paciente, con esta validación ayudamos a normalizar nuestra base de datos y no contenga datos duplicados.

Ya que el programa puede identificar estos aspectos a la vez puede verificar la cantidad de medicamentos que estamos cargando al paciente y no generar datos

redundantes. Lo último que se realiza de verificar cada registro, el programa emite una sentencia que puede ingresar los datos ya establecido, como serial del TAG, datos del paciente, habitación, diagnóstico y medicamentos, se realiza nuevamente una sentencia para el ingreso de datos en la base de datos.

```
QSqlQuery qry;
qry.prepare("INSERT INTO registro(id_tag,id_paciente,id_habitacion,diagnostico,fechaHoraIn,id_login_in) "
" VALUES ('"+id_tag+"','"+id_paciente+"','"+id_habitacion+"','"+diagnostico+"','"+tiempo+"','"+id_login+"')");
if(!qry.exec())
{
    QMessageBox::critical(this,tr("Error"),qry.lastError().text());
}
else
{
    QMessageBox::information(this,tr("Registro"),tr("Ingreso del paciente Correctamente"));
    close();
}
```

Imagen 21. Ingreso de datos para enviar a la base de datos

Para validar esta operación el sistema envía un mensaje de validación, como se muestra el código con la librería QMessageBox, que permite realizar un mensaje sin necesidad de crear un archivo .ui y solo es de carácter informativo.

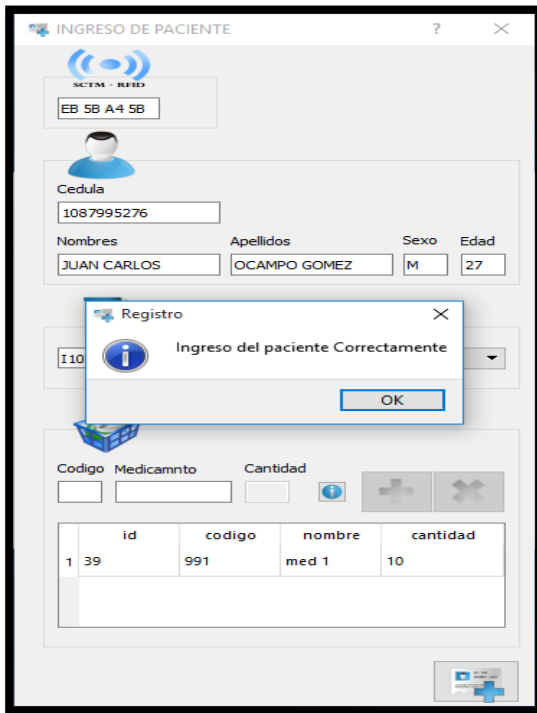


Imagen 22. Validación de ingreso paciente a la base de datos

VENTANA BUSCAR PACIENTE

Esta ventana brinda al usuario realizar la consulta del TAG y los datos que están cargados a él, con este se realiza una consulta a la base de datos, estas consultas deben ser enviadas como variables QString y generar una consulta a estos de base de datos HEIDI MYSQL, esto se logra ya que QT Creator cuenta con la librería QSqlQuery que por medio de un objeto se envía la sentencia y así extraer los datos solicitados

```
QSqlQuery qry2;  
qry2.prepare("SELECT cedula,nombre,nombre2,apellido,apellido2,sexo,DATEDIFF(curdate(),fecha_nacimiento)/365 "  
            ",diagnostico,habitacion "  
            "from registro r "  
            "inner join paciente p "  
            "on r.id_paciente=p.id_paciente "  
            "inner join habitaciones h "  
            "on r.id_habitacion=h.id_habitacion "  
            "where r.id_tag='"+id_tag+"' and fechaHoraOut is null" );
```

Imagen 23. Consulta MYSQL, condicionad por valor del serial TAG.

Reiterando que nuestro TAG es el precursor de todos los datos, con la variable id_tag, se valida en la base de datos y como muestra la siguiente figura.

The screenshot shows a window titled "Buscar Paciente" with a search bar containing "EB 5B A4 5B", a room selection dropdown set to "habitacion 1", and a medication code input field with "I10X". Below this, a patient profile is displayed with a photo icon, a name "JUAN CARLOS", and surnames "OCAMPO GOMEZ". The patient's sex is "M" and age is "27". At the bottom, a table lists medications:

	codigo	nombre	cantidad
1	991	med 1	8
2	992	med 2	5

Imagen 24. Visualización de la ventana buscar paciente, al momento de realizar la consulta a la base de datos.

Esta ventana también permite realizar modificaciones a los medicamentos cargados a los paciente, ya si necesita descargue o ingreso de medicamentos, esto se realiza con un llamado a la ventana solicitada y realizando un nuevo objeto d de la clase movimientoDialog y posteriormente ejecutándola con d.exec().

```
movimientosDialog d(this);  
d.exec();
```

Imagen 25. Objeto de la clase movimientosDialog.

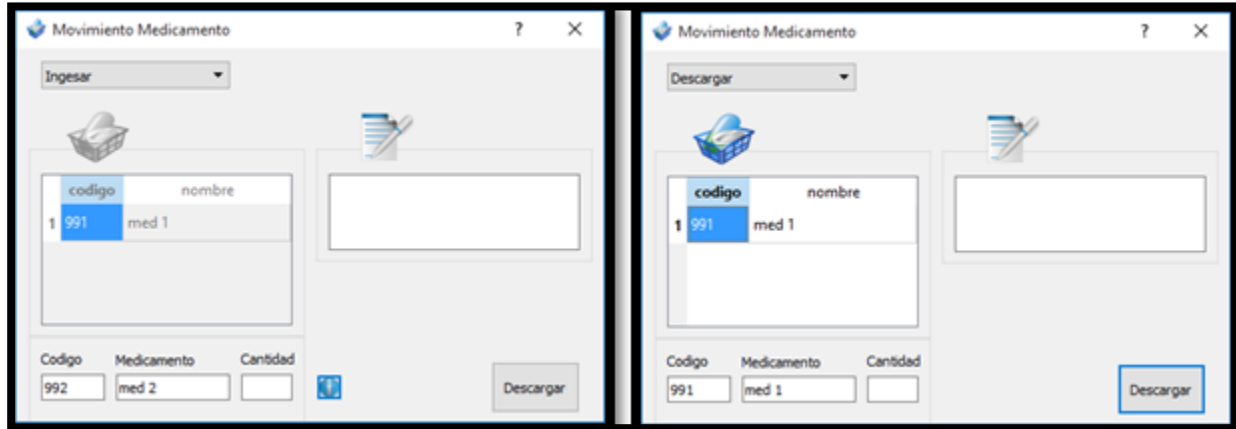


Imagen 26. Ventanas ingreso, descargue.

El programa es capaz de identificar si se realiza un descargue de medicamentos, contiene una validación de existencia del medicamento, de no tener existencias genera un aviso informativo, que cuenta con notas para escribir información del descargue.

Al ingreso del medicamento el programa es capaz de identificar si es la primera vez que se realiza el cargue o ya lo contiene, esto se realiza para evitar datos redundantes en la base de datos. Este también cuenta con una nota para escribir información del nuevo ingreso de medicamento.

VENTANA SALIDA PACIENTE

Como las anteriores ventanas ya mencionadas, esta cuenta con la validación de TAG, para traer los datos asociados de este serial. Cuenta con restricciones ya que ningún paciente se puede brindar de alta o salida, si este tiene algún medicamento cargado, es necesario realizar una nota de devolución a farmacia por el motivo que no se realizó la aplicación de dicho medicamento.

codigo	nombre	cantidad
1 991	med 1	0
2 992	med 2	0

Imagen 27. Ventana salida paciente

Se observa el objeto qry0 de la clase QSqlQuery, para realizar la consulta a la base de datos HEIDI SQL, y saber si el serial que corresponde a un paciente contiene algún medicamento no descargado.

```
QSqlQuery qry0;  
qry0.prepare("SELECT codigo,nombre,sum(cantidad_in - cantidad_out) as cantidad "  
            "from movimiento mov "  
            "inner join medicamentos m "  
            "on mov.id_medicamento=m.id_medicamento "  
            "where mov.id_paciente='"+id_paciente+"'");
```

Imagen 28. Sentencia MYSQL, consulta a base de datos

Si esta validación resulta ser negativa, brinda la posibilidad de acceder a una ventana de descargue de medicamentos por una nota de devolución

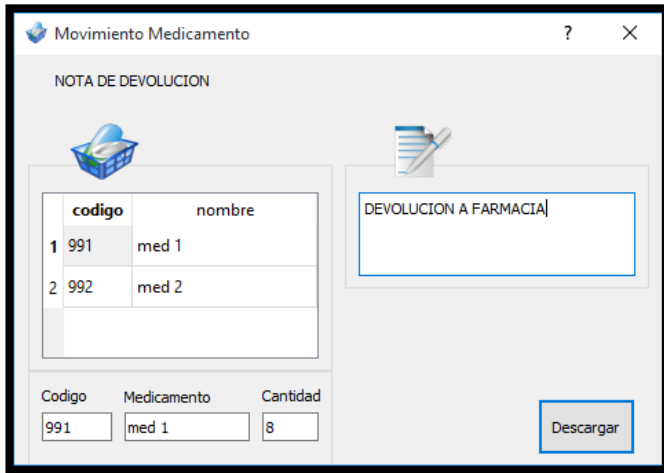


Imagen 29. Ventana nota devolución

Este cuadro está programado que todo allí será descargado y documentado por que se realiza la devolución de este medicamento, con esto realizar el descargue del medicamento de la base de datos.

VENTANA INFORMES

Esta ventana es una herramienta del programa que ayudara al manejo estadístico de la información y control de este, se maneja de 4 informes diferentes con la misma finalidad, control del medicamento, pacientes ingresados, control de usuarios.

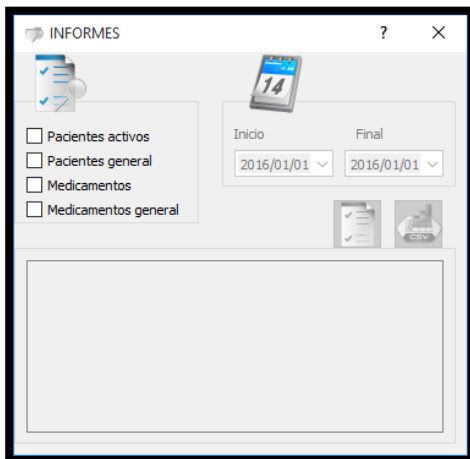


Imagen 30. Ventana Informes

Por medio de CheckBox se realiza la verificación del informe que deseamos realizar, luego de saber o indicar el informe, se habilita generar informe o validar esta información por un rango de fechas que se desee

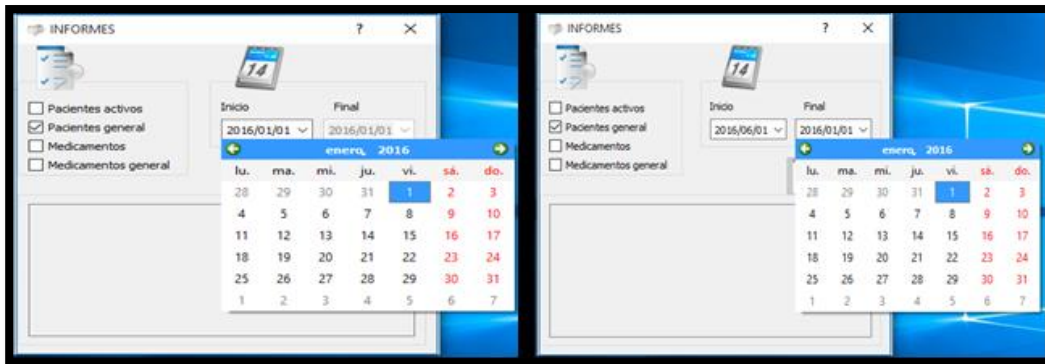


Imagen 31. Date edit para rango de fechas

Con la información que suministres el usuario se realiza la consulta que se mostrara en un Tableview, es un arreglo de datos que nos arroja la consulta por medio de la librería QSqlQueryModel, es posible realizar este procedimiento.

```

if(nombre_imforme=="Pacientes activos")
{
    ui->tableView->setEnabled(true);
    mModel = new QSqlQueryModel(this);
    mModel->setQuery("SELECT tag,cedula,nombre,nombre2,apellido,apellido2,sexo"
        ", ROUND((DATEDIFF(curdate(),fecha_nacimiento)/365),0) AS edad,habitacion,fechaHoraIn "
        "FROM registro r "
        "INNER JOIN rfidtag tg "
        "ON r.id_tag=tg.id_tag "
        "INNER JOIN paciente p "
        "ON r.id_paciente=p.id_paciente "
        "INNER JOIN habitaciones h "
        "ON r.id_habitacion=h.id_habitacion "
        "WHERE fechaHoraOut is null ");
}
    
```

Imagen 32. Sentencia MYSQL, consulta informe paciente.

Con esta esta información se puede mostrar con una tabla, en el módulo QTableView.

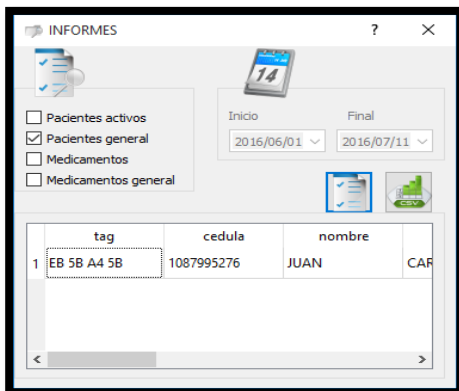


Imagen 33. Visualización de la consulta en Tableview.

Con los datos ya consultados, también tiene la opción de realizar la exportación de estos datos en archivo .CSV, este formato aplica para cualquier programa de tablas, como Excel. Para esto se necesitan librerías QFile (Creacion de archivo), QFileDialog (abrir ventana para archivo), QDir (dirección del archivo) y un arreglo de datos para que de la información realizada de la basa de datos.

```
-----guarda informe en archivo .csv-----
auto csv=QFileDialog::getSaveFileName(this,"Guardar",QDir::rootPath(),"CSV File (*.csv)");
QFile csvfile(csv);
if(csvfile.open(QIODevice::WriteOnly|QIODevice::Truncate))
{
    QTextStream out(&csvfile);
    out<<textData;
}
else
{
    QMessageBox::critical(this,tr("Error"),tr("No se puede guardar documento"));
}
csvfile.close();
```

Imagen 34. Algoritmo creación archivo .CSV

```
int rows=mModel->rowCount();
int columns=mModel->columnCount();

for (int i = 0; i < rows; i++)
{
    for (int j = 0; j < columns; j++)
    {
        textData += mModel->data(mModel->index(i,j)).toString();
        textData += ","; // separadores formato .csv
    }
    (segmentacion para cambio de linea)-----
    textData += "\n";
}
}
```

Imagen 35. Arreglo de datos para archivo.csv

Aplicación de protocolo de comunicación de entre los programas QT Creator y gestor de base de datos MARIADB (HeidiSQL).

Se decide utilizar el gestor de base de datos MARIADB derivado de MYSQL, es ejecutada por una interfaz gráfica llamada HEIDI SQL, tiene conexión TCP/IP o protocolo túnel (SSH), la conexión cliente servidor se realizada por QT Creator cuenta con una librería QSqlDataBase, es necesario activar el controlado QMYSQL este permite habilitar y realizar la conexión con HEIDI SQL, es necesarios tener los datos principales como el nombre de la base de datos a conectar, usuario, contraseña, dirección del servidor y puerto a conectar.

```
mDb = QSqlDatabase::addDatabase("QMYSQL");  
mDb.setDatabaseName(nombreBaseDatos);  
mDb.setUserName(nombreUsuario);  
mDb.setPassword(contraseyaUsuario);  
mDb.setHostName(direccionServidor);  
mDb.setPort(puertoConexion);
```

Imagen 36. Protocolo para conexión base de datos HEIDI SQL

Para nuestro caso se realiza una conexión local LOCALHOST o 127.0.0.1 para colocar en funcionamiento los servidores de HEIDI SQL.

Generar una base de datos MYSQL MARIADB (HeidiSQL) para ingreso de información de cada registro siniestrado por el TAG RFID.

Se decide utilizar el gestor de base de datos MARIADB, este es un derivado de MYSQL, es ejecutada por una interfaz gráfica llamada HEIDI SQL, que contiene la forma simplificada funcional para relazar una base de datos gratuita y completamente funcional.

Se procede a realizar una base de datos en el LOCALHOST, para garantizar los datos de esta, se procede a realizar la base de datos que en este caso la llamamos jcbasedatos como se muestra continuación

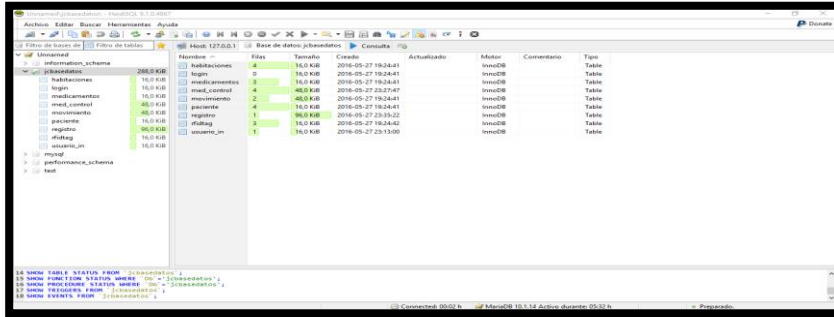


Imagen 37. Gestor de base de datos HEIDI MYSQL

Para realizar cada tabla se crea con lenguaje MYSQL, realizando el proceso de normalización y generación de valores necesarios para la funcionalidad del proyecto SCTM-RFID.

```

1 CREATE TABLE `login` (
2   `id_login` INT(11) NOT NULL AUTO_INCREMENT,
3   `usuario` CHAR(50) NULL DEFAULT NULL,
4   `contrasena` INT(11) NULL DEFAULT NULL,
5   `nombre` CHAR(50) NULL DEFAULT NULL,
6   PRIMARY KEY (`id_login`)
7 )

```

Imagen 38. Sentencia MYSQL, creación tabla LOGIN.

Para estos cumplir con el proceso del proyecto, se generan algunas tablas que solo serán modificadas por el administrador, ya que son primarias para el buen funcionamiento de este, como lo son login, medicamentos, habitaciones. Estas tablas son valores constantes que el usuario no necesita manipular.

Para ver los datos que en esta tabla se ingresa, HEIDI SQL cuenta con la visualización de estas, y si es necesario realizar su modificación desde esta.

🔑 id_login	usuario	contrasena	nombre
1	j.ocampo	123	juan ocampo
2	p.perez	123	pedro perez

Imagen 39. Tabla usuarios LOGIN

Contamos con algunas tablas con valores que nos permite realizar una restricción y validación para el sistema, como es el caso de la tabla de habitaciones

id_habitacion	habitacion	estado
1		in
2	habitacion 1	in
3	habitacion 2	in
4	habitacion 3	in

Imagen 40. Tabla Habitaciones con variables estática

Esta tiene una columna primaria, habitación y por ultimo su estado “in” o “out”, al momento de estar ocupada o disponible para la asignación a un serial TAG.

También se realiza llaves foráneas de cada una de las tablas para construcción de un registro e implementación dicho de la base de datos. Estas llaves foráneas nos ayudan a normalizar la base de datos y a realizar que los datos no se dupliquen.

id_med_control	id_mov_med	tipo	cantidad	fecha_hora	notaDevolucion	id_login
88	39	ingreso	10	2016-06-11 11:09:43	primer ingreso	1
90	39	Descargar	2	2016-06-11 11:16:03	PRIMERA DOSIS	1
91	41	ingreso	5	2016-06-11 11:16:03	INGRESO MEDICAMENTO	1
92	39	Nota de devolucion	8	2016-06-11 11:23:47	DEVOLUCION A FARMACIA	1
93	41	Nota de devolucion	5	2016-06-11 11:24:32	DEVOLUCION A FARMACIA	1
94	42	ingreso	2	2016-06-12 16:36:43	primer ingreso	1
95	42	Nota de devolucion	2	2016-06-12 16:37:31	salida	1
96	43	ingreso	5	2016-06-12 16:37:53	primer ingreso	1
97	43	Descargar	2	2016-06-12 16:38:58	descargar	1
98	43	Descargar	3	2016-06-12 16:38:58	descargar	1
99	44	ingreso	2	2016-06-12 16:39:41	ingresar	1
100	44	Nota de devolucion	2	2016-06-12 16:40:17	salida	1

Imagen 41. Tabla de registros

PRESUPUESTO Y FUENTES DE FINANCIACIÓN

El presupuesto para la realización del proyecto control y trazabilidad de medicamentos con tecnología RFID es el siguiente:

Concepto	Cantidad	Valor Unitario	Valor Total
Computador portátil	1	\$1.000.000	\$1.000.000
Tarjeta ARDUINO	2	\$72.000	\$144.000
Etiquetas RFID	10	\$3.300	\$33.000
Módulo RFID-RC522	2	\$150.000	\$300.000
Licenciamiento de Software (ARDUINO,QT Creator y MARIADB (HeidiSQL))	1	Software gratuito.	\$0
Otros	1	\$150.000	\$150.000
Total			\$ 1.627.000

Tabla 3. Tabla costo del proyecto

La fuente de financiación la suministrará la empresa:

ELECTROTICS Spin Off, Centro De Innovación Y Desarrollo Tecnológico de la Universidad Tecnológica de Pereira, CEO Ingeniero Electrónico Marcelo Bernal Gómez.

CONCLUSIONES

La implementación de un sistema de control y trazabilidad de medicamentos con la tecnología RFID, fue exitosa, reflejando la culminación del programa con sus múltiples funcionalidades y requerimientos que se planteó en los objetivos, es un punto de partida para ser escalable a plataformas móviles como ANDROID y acceder al programa desde cualquier dispositivo.

Se logra realizar una implementación óptima en la base de datos HEIDI SQL, normalizándola y posteriormente ser escalable si es necesario implementa nuevos datos.

Este proyecto tenía planeado realizar las pruebas de funcionamiento en el Hospital San Jorge en el área de oncología, lastimosamente las directivas del hospital no brindaron la autorización para realizar las pruebas al programa SCTM-RFID, informando que el hospital se encuentra en reestructuración y remodelaciones que impedían esta solicitud.

Para revisar la aceptación del SCTM-RFID se muestra su funcionamiento a personal del área de la salud y realizar pequeños simulacros de funcionalidad, aprobando su practicidad y funcionamiento. Para la implementación a futuro en una IPS.

Se resalta la importancia de implementar en SCTM-RFID de hardware de bajo costo y software libre, para lograr importantes avances en el campo tecnológico con un bajo presupuesto, y de ser un punto de partida para posteriores estudios e implementaciones que en la actualidad.

BIBLIOGRAFIA

Colombia, G. (s.f.). *gs1co.org*. Obtenido de <http://www.gs1co.org/>

Fernández, S. A. (s.f.). *RFID: La tecnología*. Recuperado el 25 de Noviembre de 2015
rfidjournal. (s.f.). *rfidjournal.com*. Recuperado el 25 de enero de 2016, de
<http://www.rfidjournal.com/articles/view?1339/>

Sabana, U. d. (s.f.). *inalde.edu.co*. Recuperado el 25 de Enero de 2016, de
<http://www.inalde.edu.co/sala-de-prensa/revista-inalde/detalle-blog/ic/atencion-segura-y-decalidad-al-paciente-a-traves-del-manejo-del-proceso-hospitalario/icac/show/Content/>

Social, M. d. (s.f.). *minsalud.gov.co*. Recuperado el 20 de Noviembre de 2015, de
<https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/CA/seguridad-en-la-utilizacion-de->

[medicamentos.pdf#search=%252CMEJORAR%2520LA%2520SEGURIDAD%2520EN%2520LA%2520UTILIZACION%2520DE%2520MEDICAMENTOS](#)

(mepsrealtime, Intelliguard® RFID Solutions, 2016)

(Arduiono, 2016)

(Eirik, 2016)

(Corporation, 2016)