

Diseño del sistema:

Introducción:

El sistema constará de dos bloques principales, uno que se realizará en Visual Studio y otro en Arduino, que ejecutarán el software y el hardware respectivamente.

- Estructura general de la aplicación que se ejecuta en el PC:

```
#include <stdio.h>
```

```
#include <malloc.h>
```

```
Struct paciente {
```

```
    Char [31] nombre, sintomas[201];
```

```
    Int edad, habitacion, ingreso;
```

```
};
```

```
int main() {
```

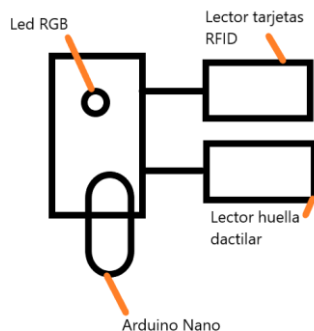
```
// Conectar con el arduino (Lector tarjeta rfid, lector huella dactilar)
```

```
// Contraseña del usuario (char[31] usuario, char contraseña), si el usuario y la contraseña son correctos, el código continúa, si no queda en bucle
```

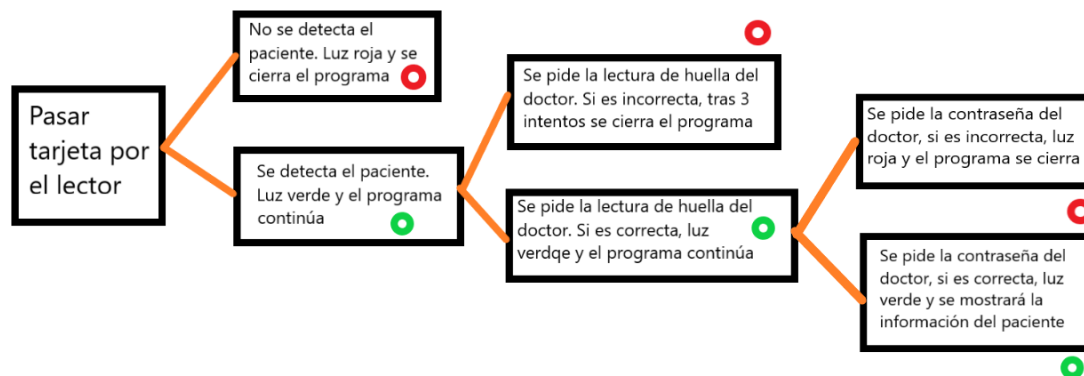
```
// Incorporación de datos por teclado: Sustitución y adición de cadenas de síntomas
```

```
}
```

Estructura general de la aplicación que se ejecuta en la plataforma hardware:



Estructura general del código:



Funcionalidades del sistema:

1. Lectura sensor RFID:

Este será un código en Arduino, será necesario para que nuestro lector RFID pueda leer el código de identificación de cada tarjeta.

```

#include <SPI.h>#include <MFRC522.h>#define RST_PIN9 //Pin 9 para el reset del RC522#define
SS_PIN10 //Pin 10 para el SS (SDA) del RC522MFRC522 mfrc522(SS_PIN, RST_PIN); //Creamos el
objeto para el RC522void setup() {Serial.begin(9600); //Iniciamos la comunicación serialSPI.begin();
//Iniciamos el Bus SPImfrc522.PCD_Init(); // Iniciamos el MFRC522Serial.println("Lectura del
UID");}void loop() { // Revisamos si hay nuevas tarjetas presentesif (mfrc522.PICC_IsNewCardPresent
()) { //Seleccionamos una tarjeta if (mfrc522.PICC_ReadCardSerial()) { //
Enviamos serialemente su UID Serial.print("Card UID:"); for (byte i = 0; i <
mfrc522.uid.size; i++) { Serial.print(mfrc522.uid.uidByte[i] < 0x10? " 0": " ");
Serial.print(mfrc522.uid.uidByte[i], HEX); } Serial.println(); //
Terminamos la lectura de la tarjeta actual mfrc522.PICC_HaltA (); } }}
  
```

2. Comunicación Arduino - visual:

Una vez leído el código de identificación de la tarjeta, será necesario conectar el Arduino con nuestro programa de visual studio. Esto será necesario para enviar la información del código de identificación de la tarjeta a nuestro software y también para que nuestro led rgb pueda funcionar acorde con la introducción de la contraseña del médico y también que nuestro código en visual pueda detectar cuándo ha funcionado el lector de huella para continuar con el programa.

3. Identificación del médico:

```

char usuario1[ ] = 'pablo.doctor@hospital.es', usuario2[ ] = 'javier.doctor@hospital.es', usuario3[ ] =
'jaime.doctor@hospital.es', contraseña1[ ]='doctorvandenbos', contraseña2[ ]='capricornio420',
contrasña3 [ ]= 'doctortorresabad', usuario[31], contraseña[31];
  
```

```
int i=0;
```

```

for(i=5;i=0;i--)

gets_s usuario;

If(usuario==usuario1)

    {

gets contraseña;

If (contraseña==contraseña1)

{

printf ("Bienvenido %s", usuario1);

        }

    else

        printf("La contraseña es incorrecta, tiene %d intentos mas", i);

        i--;

    }

else if (usuario==usuario2)

    {

gets contraseña;

If (contraseña==contraseña2)

{

printf ("Bienvenido %s", usuario2);

        }

    else{

        printf("La contraseña es incorrecta, tiene %d intentos más", i);

        i--;

    }

```

Haremos un if else anidado para ser capaces de introducir un nombre de usuario y una contraseña en concreto para cada doctor.

4. Análisis de datos 'struct'

```
Struct paciente [ N] {  
    Char [31] nombre, sintomas[201];  
    Int edad, habitacion, ingreso;  
};
```

Una vez que tengamos las tarjetas rfid, crearemos un struct para cada paciente, siendo N el número de tarjetas de las que dispongamos (pacientes)

5. Visualización en pantalla:

La interacción entre el usuario y el programa se realizará en visual studio. El hardware será un apoyo para controlar la identificación del usuario y mejorar la seguridad del sistema. Para modificar los datos de la variable struct de cada paciente, optimizaremos el programa para que el médico en cuestión pueda sustituir cadenas de caracteres, añadir cadenas o eliminar cadenas.

Hemos realizado un pequeño programa en arduino para trabajar con un led rgb. De momento no tiene las indicaciones que queremos, pero a partir de este programa tenemos una base para empezar a trabajar con este led.

```
const int ledPinRed = 3;  
  
const int ledPinGreen = 6;  
  
const int ledPinBlue = 5;  
  
void setup() {  
    // put your setup code here, to run once:  
    Serial.begin(9600);  
    pinMode(ledPinRed, OUTPUT);  
    pinMode(ledPinGreen, OUTPUT);  
    pinMode(ledPinBlue, OUTPUT);  
}  
  
void setLedColor(byte red, byte green, byte blue) {  
    analogWrite(ledPinRed, red);  
    analogWrite(ledPinGreen, green);  
    analogWrite(ledPinBlue, blue);  
}
```

```

void loop() {

  // put your main code here, to run repeatedly:

  setLedColor(255, 255, 0);

  delay(500);

  setLedColor(0, 0, 255);

  delay (200);

}

```

Estrategias de monitorización:

