

Universidad Autónoma Metropolitana
Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Sistema móvil para alertas de consumo de medicamentos y seguimiento de
pacientes con diabetes

Proyecto Tecnológico

Trimestre 2018 Otoño

Laura Fernanda Eizmendi Hernández

2122003812

al2122003812@azc.uam.mx

M. en C. Josué Figueroa González

Profesor Asociado

Departamento de Sistemas

jfgo@correo.azc.uam.mx

11 de enero de 2019

Declaratoria

A handwritten signature in blue ink, appearing to read "Josué F. G.", with a stylized flourish at the end.

Yo, M. en C. Josué Figueroa González, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

A handwritten signature in blue ink, appearing to read "Laura F. E.", with a stylized flourish at the end.

Yo, Laura Fernanda Eizmendi Hernández, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Resumen

Esta aplicación está dedicada para los pacientes diabéticos y tiene la principal finalidad de llevar un mejor manejo y control de la enfermedad.

Las funcionalidades que puede encontrar el paciente son:

- Registro de sus datos personales.
- Registro de un tutor por medio de un nombre y número celular.
- Registro de medicamentos, el inicio de toma y la hora de administración.
- Registro de citas médicas, se visualiza el nombre de la cita, la especialidad de la rama médica, se despliega un calendario y la hora del recordatorio.
- Registro de glucosa, permite saber al paciente las medidas que ha tomado por medio de una fecha que se despliega con un calendario.

Tabla de contenido

Tema	Página
Declaratoria.....	1
Resumen.....	2
Introducción.....	5
Antecedentes.....	6
Justificación.....	7
Objetivos.....	8
Marco teórico.....	9
Desarrollo del proyecto.....	12
Resultados.....	23
Conclusiones.....	24
Anexo A. Código fuente.....	25
Bibliografía.....	45

Índice de imágenes

Imagen	Página
Figura 1. Entorno de desarrollo.....	10
Figura 2. Diagrama de casos de uso.....	12
Figura 3. Menú principal.....	12
Figura 4. Pantalla del registro de paciente A.....	13
Figura 5. Pantalla del registro de paciente B.....	13
Figura 6. Pantalla de registro de medicamento.....	14
Figura 7. Pantalla de selección de fecha.....	15
Figura 8. Pantalla de selección de la hora.....	15
Figura 9. Panel de notificaciones principales del celular.....	15
Figura 10. Pantalla de selección de la hora para citas médicas.....	16
Figura 11. Pantalla de selección de la fecha para citas médicas.	16
Figura 12. Panel de notificaciones principales del celular.....	16
Figura 13. Pantalla de registro de tutor.....	17
Figura 14. Pantalla de registro de glucosa.....	17
Figura 15. Pantalla de registro de la fecha correspondiente a glucosa....	17
Figura 16. Diagrama entidad relación.....	18

Figura 17. Caso de uso: Registrar paciente.....	19
Figura 18. Caso de uso: Registrar tutor.....	19
Figura 19. Caso de uso: Registrar medicamento.....	20
Figura 20. Caso de uso: Registrar cita médica.....	20
Figura 21. Caso de uso: Registrar glucosa.....	21
Figura 22. Diagrama UML.....	22

Introducción

La diabetes es la causa principal de muerte en México ya que es una enfermedad no curable, pero afortunadamente una vez diagnosticada es controlable a partir del consumo oportuno de medicamentos y seguimiento de tratamientos. Uno de los principales problemas, es que, en muchas ocasiones, los diabéticos no los consumen a las horas indicadas y dosis correctas lo que les ocasiona que tengan crisis.

Actualmente, con la aparición de las aplicaciones móviles muchas personas han cambiado la manera de consultar información o llevar el control de sus actividades. En el mercado las aplicaciones móviles han tenido una respuesta favorable del público, ya que por economía o quizás por espacio, se les hace más sencillo visualizar o consultar una aplicación desde un dispositivo móvil, además las aplicaciones móviles tienen una amplia difusión de los contenidos que se desea compartir, así como la facilidad y velocidad para el acceso a la información.

Considerando la importancia del correcto consumo de medicamentos para la diabetes y las ventajas que ofrecen las aplicaciones móviles, el sistema de control y tratamiento de la diabetes se ejecutará en un dispositivo móvil que permitirá que a un paciente se le pueda informar del momento en que debe tomar sus medicamentos y enviar alertas a sus familiares si no lo ha hecho además de permitir a su médico conocer esta información para dar mejor seguimiento al estado del diabético.

Antecedentes

“Sistema de Apoyo Clínico para el Tratamiento del Paciente Diabético” [1]: El sistema permite gestionar el expediente clínico, guarda información del paciente, así como sus análisis, dietas y reportes gráficos de cada diabético. Teniendo la facilidad de obtener el reporte del paciente cuando el doctor lo requiera. Está relacionado con el sistema móvil, sin embargo, es especializado para el uso exclusivo del médico, es un sistema web y no envía alertas para un mejor control de la enfermedad.

“Reingeniería del Sistema de Consulta y Diagnóstico médico veterinario SICODIVE” [2]: Es un sistema web que apoya al médico en la gestión de consultas médicas, diagnóstico de pacientes, gestión de expedientes, usuarios y citas. Este sistema se relaciona de manera estrecha con el sistema móvil diabético, sin embargo, está especializado a médicos veterinarios.

“Prototipo de una aplicación móvil para la gestión de síntomas de un paciente” [3]. Está enfocado en editar, modificar, eliminar y obtener resumen de los síntomas, pruebas en los módulos de respaldo y recuperación de síntomas de un paciente. La principal diferencia es que este sistema está particularmente especializado en los síntomas del paciente, sin llevar a cabo un control sus medicamentos.

“Optimizador Control Diabético” [4]: Es una aplicación móvil que permite el control de la enfermedad por medio un registro del nivel de Glucemia, y una barra que, dependiendo del nivel de glucosa, se va recomendando una dieta específica para un nivel óptimo de azúcar en la sangre. La principal diferencia es que este sistema no lleva un seguimiento de medicamentos, simplemente muestra niveles de glucosa.

“Diabetes, Control y Manejo” [5]: Aplicación móvil que ayuda al paciente a manejar la enfermedad por medio de fases, desde un cambio de mentalidad, hasta el control de triglicéridos, y audios para el control, relajación de esta enfermedad. En diferencia con el sistema, es que basa el control de la diabetes mediante dietas, no con medicamentos.

“BG Monitor” [6]: Aplicación móvil que permite el registro de insulinas y medicamentos para el paciente diabético, la principal diferencia es que no necesita un registro previo del paciente, no agenda citas médicas, y no manda alertas a un tutor, en caso de no tomar el medicamento, para mayor seguimiento del diabético en su tratamiento.

Justificación

Muchos pacientes no dan un seguimiento adecuado a sus tratamientos, lo que provoca que su estado de salud empeore. La diabetes es una enfermedad que necesita ser controlada a partir de un adecuado consumo de medicamentos, por esta razón, se propone un sistema móvil que dará la facilidad de registrar medicamentos, horas de suministro, calendario para citas médicas y un tutor (médico o familiar) en caso de que el paciente no responda las alertas que envía el sistema para la ingesta de medicinas.

Objetivo General

Diseñar e implementar una aplicación móvil para generar alertas sobre el consumo de medicamentos y dar seguimiento a pacientes con diabetes.

Objetivos Específicos

- Programar un módulo que permita registrar los datos personales del paciente
- Programar un módulo que permita registrar medicamentos y sus horarios respectivos de cada uno.
- Diseñar e implementar un calendario que permita al paciente registrar citas médicas
- Programar un módulo que permita registrar un tutor.
- Programar e implementar un módulo que, por medio de una alerta, permita el correcto consumo del medicamento.
- Programar un módulo que permitirá llevar el control de contenido de medicamentos.
- Programar un módulo en caso de no tener respuesta por el paciente diabético, se enviará una alerta al tutor para que evalúe la salud del paciente.
- Programar un módulo para el registro de toma de glucosa en la sangre, este permitirá visualizar una gráfica con las diferentes muestras por día.

Marco Teórico

Para el impulso de este proyecto fue necesario contar con un entorno de desarrollo integrado llamado Android Studio.

Razones por las cuales se utilizó el IDE:

- Tiene un potente editor de código, ejemplo: autocompletado.
- Existe la efectividad de las aplicaciones aumentando su productividad.
- Cuenta con una gran variedad de dispositivos virtuales para probar las aplicaciones gracias al emulador.
- Utiliza la herramienta de GitHub para importar algunos ejemplos de códigos.
- Mide el rendimiento de las aplicaciones utilizando las herramientas Lint.
- Es el IDE oficial para el desarrollo de aplicaciones Android.

Estructura del proyecto.

El entorno de desarrollo es amigable para el usuario ya que divide el proyecto en módulos, ejemplo: **Módulos de apps para Android, Módulos de bibliotecas, Módulos de app engine** (soporte incorporado para Google Cloud Platform). La división de los módulos facilita el acceso a los archivos del proyecto.

Cada módulo de una aplicación contiene las siguientes carpetas:

- **Manifest:** AndroidManifest.xml → Archivo de configuraciones básicas de la aplicación.
- **Java:** Contiene todos los archivos fuente. *Nota: También trabaja con el lenguaje kotlin
- **Res:** Contiene todos los recursos necesarios para que funcione la aplicación correctamente, ejemplo: XML, mapa de bits, etc.

Ventana principal de Android Studio.

La ventana principal que se muestra en la Figura 1, consta de diversas funciones:

- Barra de herramientas, permite correr el proyecto, guardar, etc.
- Barra de navegación, es una vista compacta del proyecto.
- Barra de estado, muestra algún mensaje o advertencia de la aplicación.
- Ventana de herramientas, permite acceder a acciones específicas del proyecto.
- Ventana de editor, donde se modifica o crea el código de la aplicación.

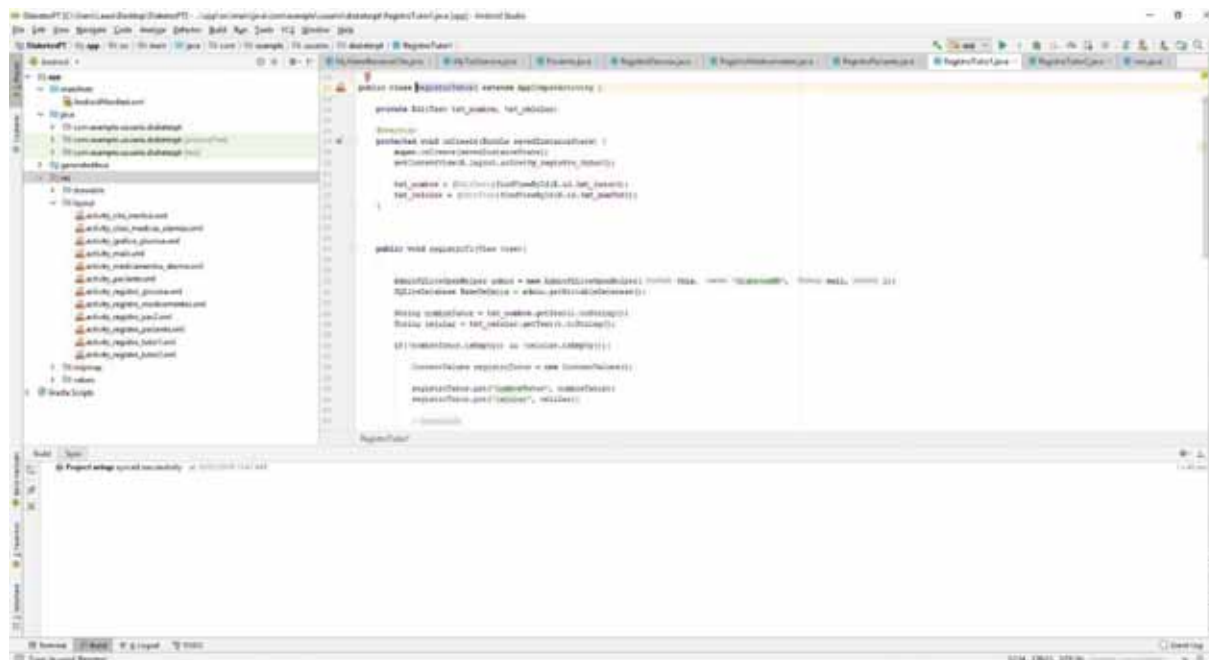


Figura 1. Entorno de desarrollo

Para este proyecto se utilizó el lenguaje orientado a objetos Java, por la razón que Android Studio soporta todas las funcionalidades de Java 7, además de que el lenguaje es independiente a la plataforma, es adaptable a todo tipo de dispositivos móviles.

Bibliotecas utilizadas para el desarrollo del proyecto:

- SQLiteOpenHelper: Es una biblioteca que permite manipular de manera fácil una base de datos alojada en un dispositivo móvil, si se requiere crear una clase descendiente ella contará con dos métodos importantes: onCreate () y onUpgrade (). Los métodos de esta clase son getReadableDataBase() y getWritableDatabase() que abren la base de datos en modo lectura y escritura.
- SQLiteDatabase: Esta biblioteca contiene métodos para crear, eliminar, ejecutar comandos SQL y realizar tareas de administración en una base de datos.
- AlarmManager: Proporciona servicios de alarma del sistema, programa en la aplicación para que la alarma suene en algún momento determinado.
- BroadcastReceiver: Implementa una política de bloqueo para garantizar que el teléfono siga sonando y funcione con normalidad.
- TimePickerDialog: Despliega un cuadro para que el usuario solicite la hora que desea.
- DatePickerDialog: Despliega un cuadro para que el usuario solicite la fecha que desea..
- NotificationManager: Es una clase que notifica al usuario eventos que suceden.

- RingtoneManager: proporciona acceso a tonos de llamada, notificaciones y otros tipos de sonidos.
- IntentService: Clase que maneja solicitudes asíncronas.
- Toast: Biblioteca que permite mostrar un mensaje en pantalla durante unos segundos al usuario para luego volver a desaparecer automáticamente sin requerir ningún tipo de actuación por su parte.

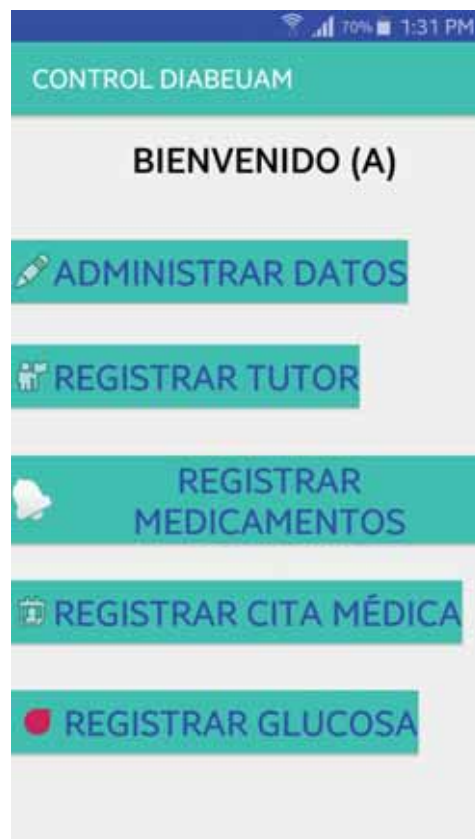
Desarrollo del proyecto

Para el desarrollo de este proyecto se utilizó un diagrama de “Casos de uso”, se muestra en la Figura 2 que nos permite visualizar lo que puede hacer el usuario con la aplicación.



Figura 2. Diagrama de casos de uso

Menú principal: En la Figura 3 se muestra la pantalla permite visualizar al paciente las principales opciones que le brinda la aplicación.



Módulo de registro del paciente: En la Figura 4 y Figura 5 se muestra el módulo que permitirá al paciente registrar sus datos personales, como es el nombre, apellido, edad, sexo, correo electrónico, así como el tipo de diabetes que padece.

CONTROL DIABEUAM

Registro del paciente

Nombre _____

Apellido _____

Edad _____

Correo electrónico _____

Sexo

Masculino
 Femenino

Tipo de diabetes

Figura 4. Pantalla del registro de paciente
A

CONTROL DIABEUAM

Apellido _____

Edad _____

Correo electrónico _____

Sexo

Masculino
 Femenino

Tipo de diabetes

Tipo 1
 Tipo 2

REGISTRAR

Figura 5. Pantalla de registro de paciente
B

Módulo de registro de medicamentos y alerta: En las Figura 6, Figura 7, Figura 8 y Figura 9, se muestra el módulo que permitirá al paciente registrar el nombre de los medicamentos, la fecha en la que inicia el medicamento y horarios de consumo. Una vez registrados, la aplicación alertará al paciente sobre su medicamento.

CONTROL DIABEUAM

Medicamento

Mensaje

FECHA

1-10-2019

HORA

hh/mm

GUARDAR

Figura 6. Pantalla de registro de medicamento



Figura 7. Pantalla de selección de fecha.
Lo toma directamente del sistema

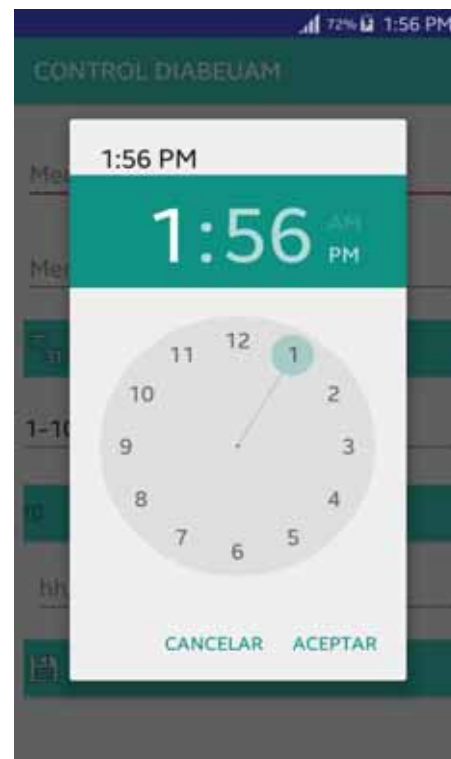


Figura 8. Pantalla de selección de la hora.
Es tomada del sistema

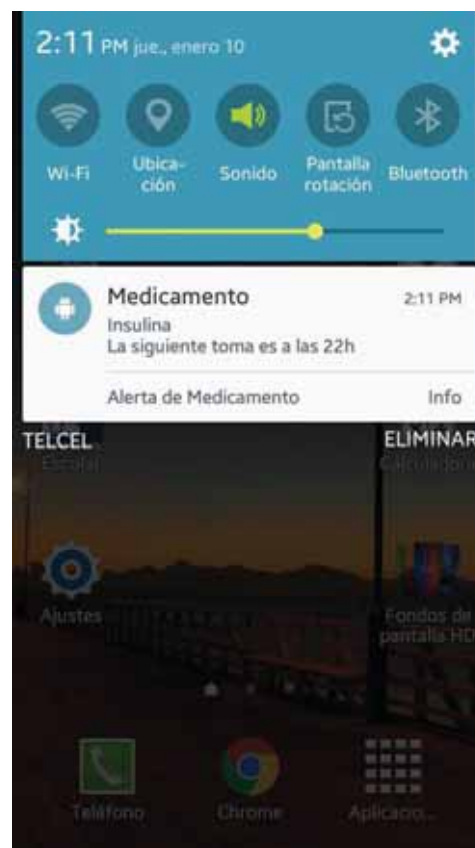


Figura 9. Panel de notificaciones principales del celular

Módulo de registro de citas médicas y alerta: En la Figura 10, Figura 11, Figura 12, permitirá registrar la fecha Figura 8 y hora de citas médicas. Una vez registrada, la alarma sonará, evitando que el paciente olvide el seguimiento de la enfermedad con su médico.

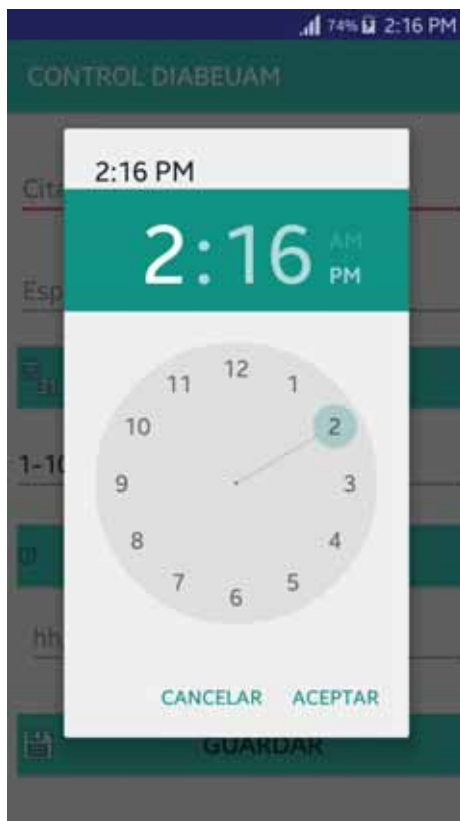


Figura 10. Pantalla de selección de la hora. para citas médicas Es tomada del sistema



Figura 11. Pantalla de selección de la fecha. para citas médicas. Es tomada del sistema



Módulo de registro de un tutor: En la Figura 13 El paciente registrará el nombre y número telefónico de dos tutores, ya sean familiares o al gusto del paciente.

CONTROL DIABEUAM

Registro Tutor

Nombre

Celular

REGISTRAR

MENÚ PRINCIPAL

Figura 13. Pantalla de registro de tutor

Módulo de registro tomas de glucosa: En la Figura 14 y Figura 15 el módulo registrará el nivel de glucosa.

CONTROL DIABEUAM

Registro de Glucosa

Glucosa mg/dl

FECHA dd/mm/aa

1 2 3

4 5 6 Sig.

7 8 9 .

0

Figura 14. Pantalla de registro de glucosa

CONTROL DIABEUAM

jueves

ENE.

10

2019

enero de 2019

D	L	M	M	J	V	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

febrero de 2019

CANCELAR ACEPTAR

Figura 15. Pantalla de registro de la fecha correspondiente a la

Diagrama entidad relación de la base de datos.

Para el almacenamiento de los datos, se utilizó la herramienta de Software libre SQLite, por su eficiencia, rapidez y sencillo manejo de consultas a celulares con poca capacidad, el diagrama que se utilizó fue el siguiente:

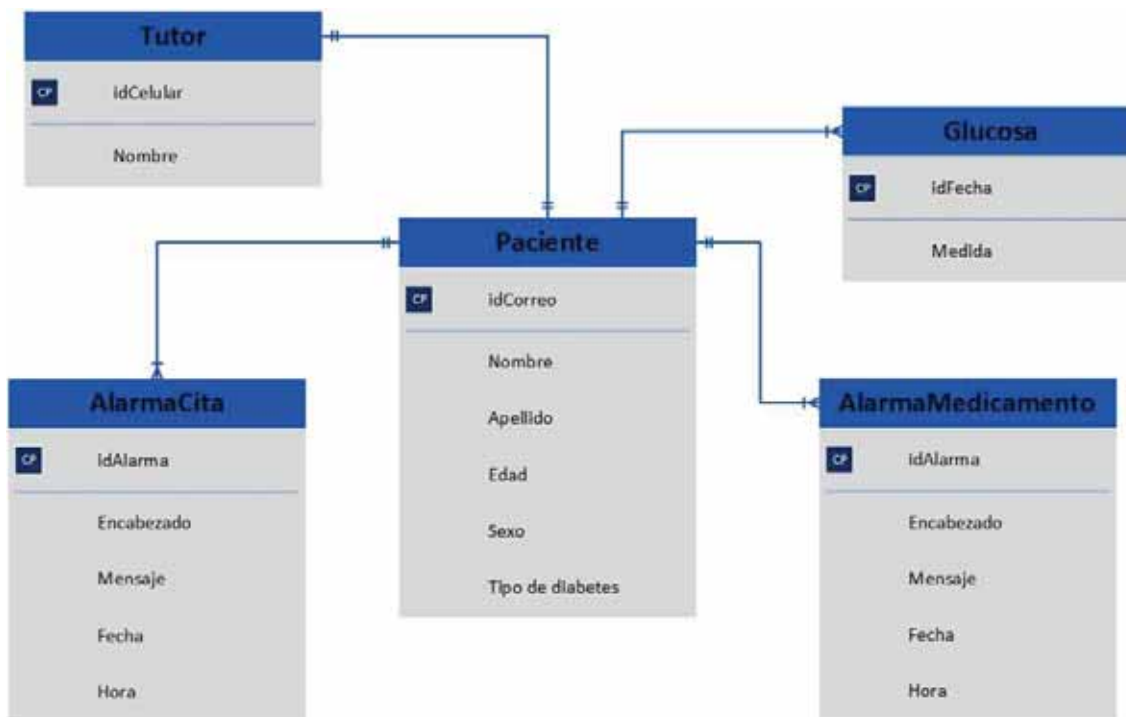


Figura 16. Diagrama entidad relación.

Diccionario de datos:

- Tabla paciente: Incluye todos los datos personales del paciente, como su edad, correo, nombre, apellido, sexo y el tipo de diabetes que padece.
- Tabla Tutor: Contiene solo dos campos, que es el nombre y número celular.
- Tabla AlarmaCita: Contiene los atributos idAlarma, que será única, un encabezado que define el contexto de la cita, el mensaje para tener un conocimiento previo sobre que especialidad es la cita, la fecha y hora de dicho encuentro médico.
- Tabla AlarmaMedicamento: Contiene los atributos idAlarma, que será única, un encabezado que define el contexto del medicamento, el mensaje para tener un conocimiento previo a qué hora será la siguiente toma, la fecha y hora de dicha toma de la dosis.
- Tabla Glucosa: Contiene dos atributos, el idFecha, y la medida de la glucosa que se mide por g/mol.

Diagrama de secuencia por cada caso de uso

Caso de uso: registrar paciente.

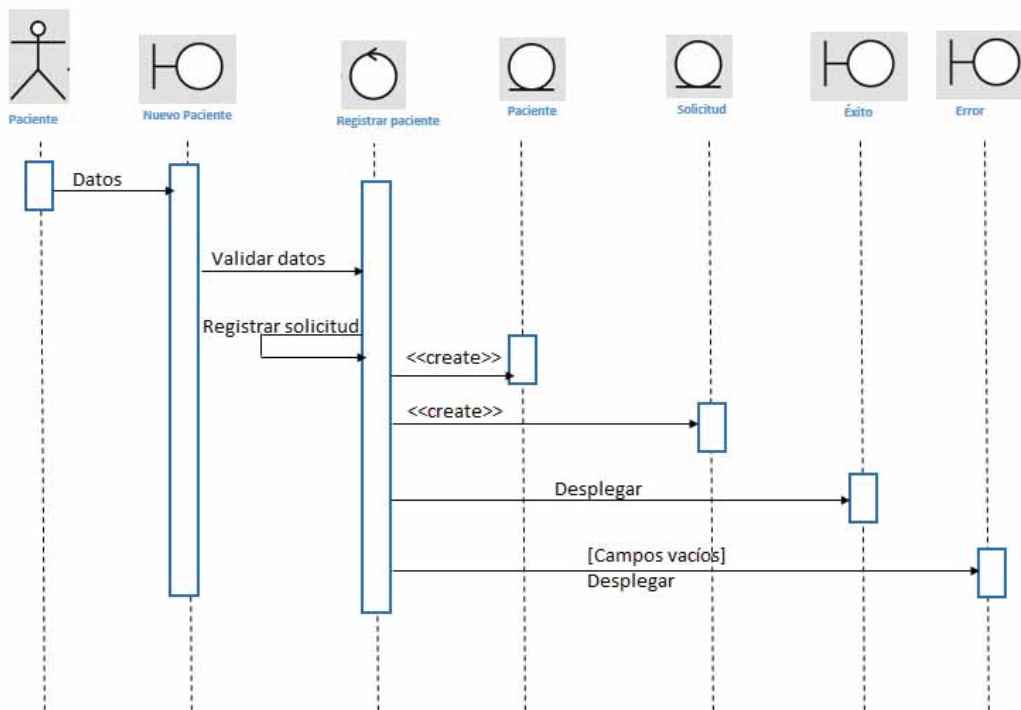


Figura 17. Caso de uso: Registrar paciente.

Caso de uso: registrar tutor

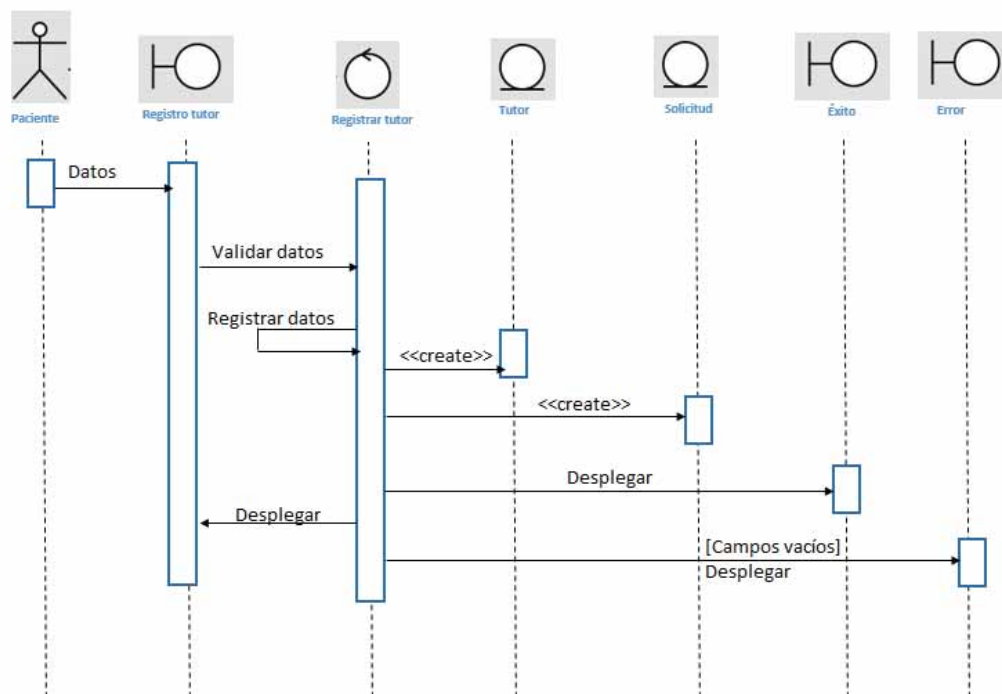


Figura 18. Caso de uso: Registrar tutor.

Caso de uso: Registrar medicamento

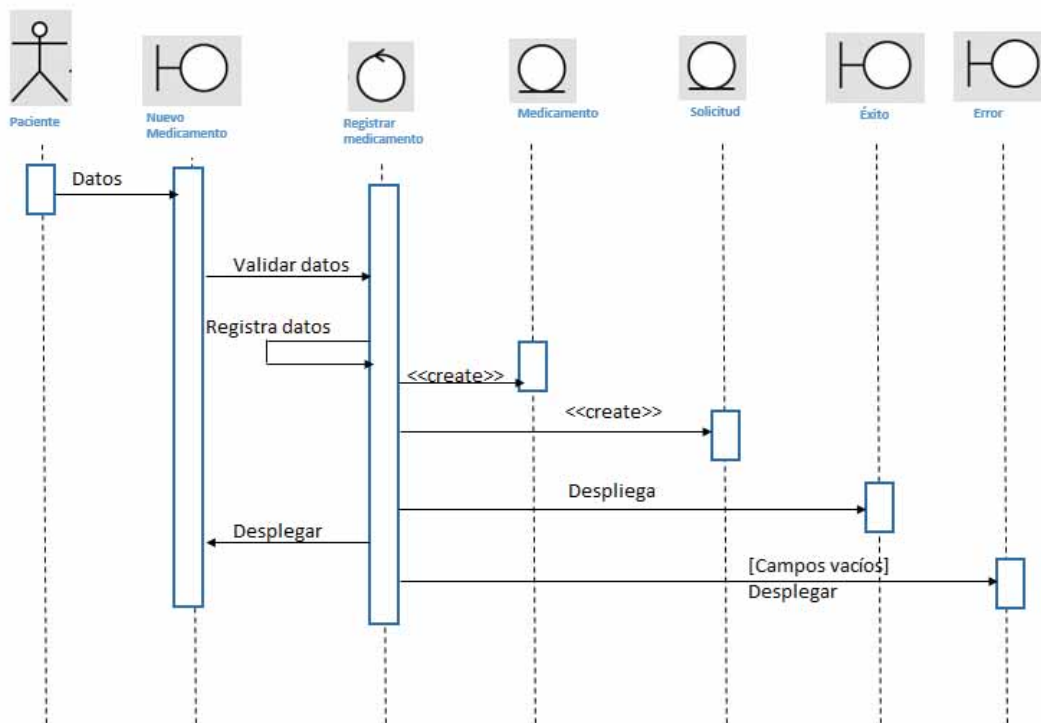


Figura 19. Caso de uso: Registrar medicamento.

Caso de uso: Registrar cita médica

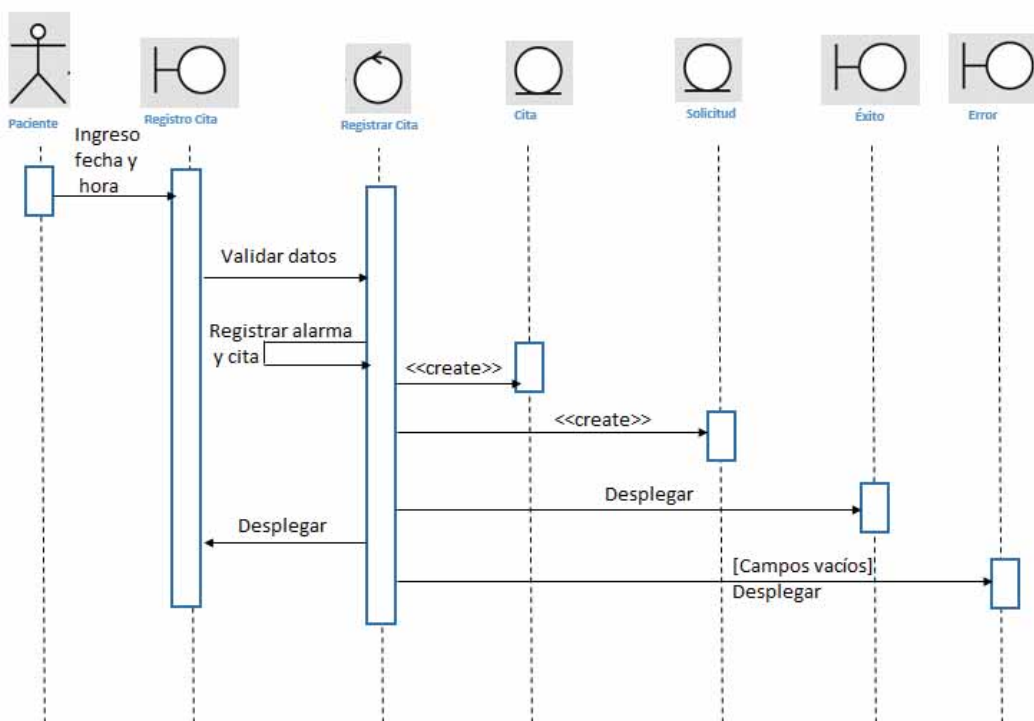


Figura 20. Caso de uso: Registrar cita médica.

Caso de uso: Registrar glucosa

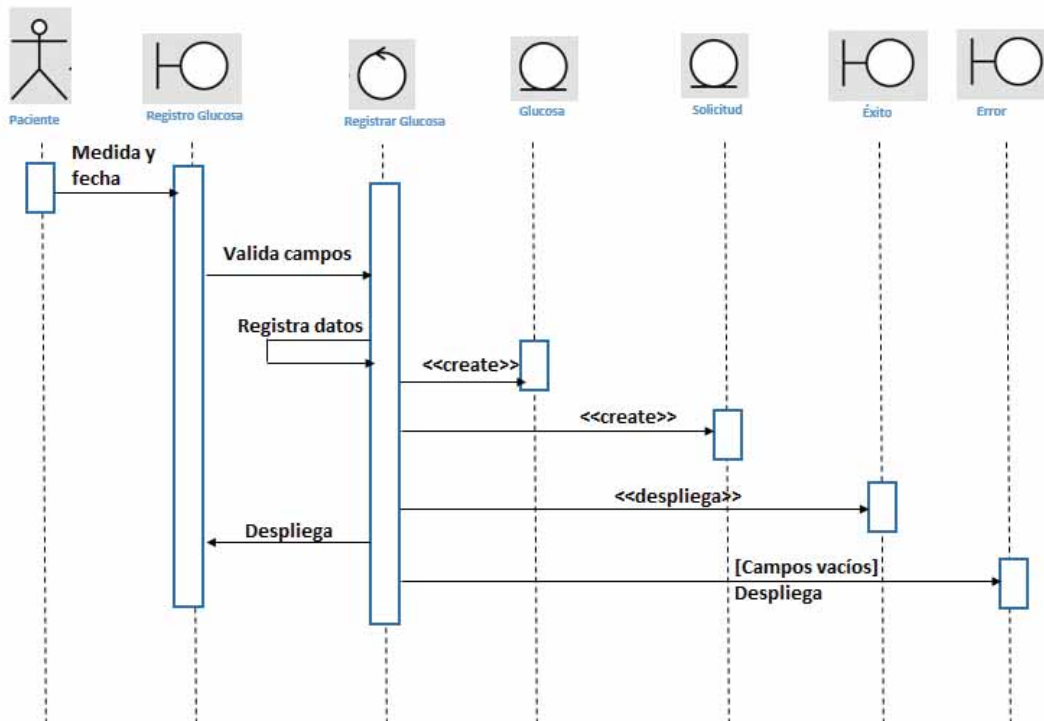


Figura 21. Caso de uso: Registrar glucosa.

Diagrama UML

Para una mejor comprensión del proyecto se realizó un diagrama UML que muestra la interacción entre las clases, los atributos y los métodos utilizados para el desempeño correcto de la aplicación. Figura 21.

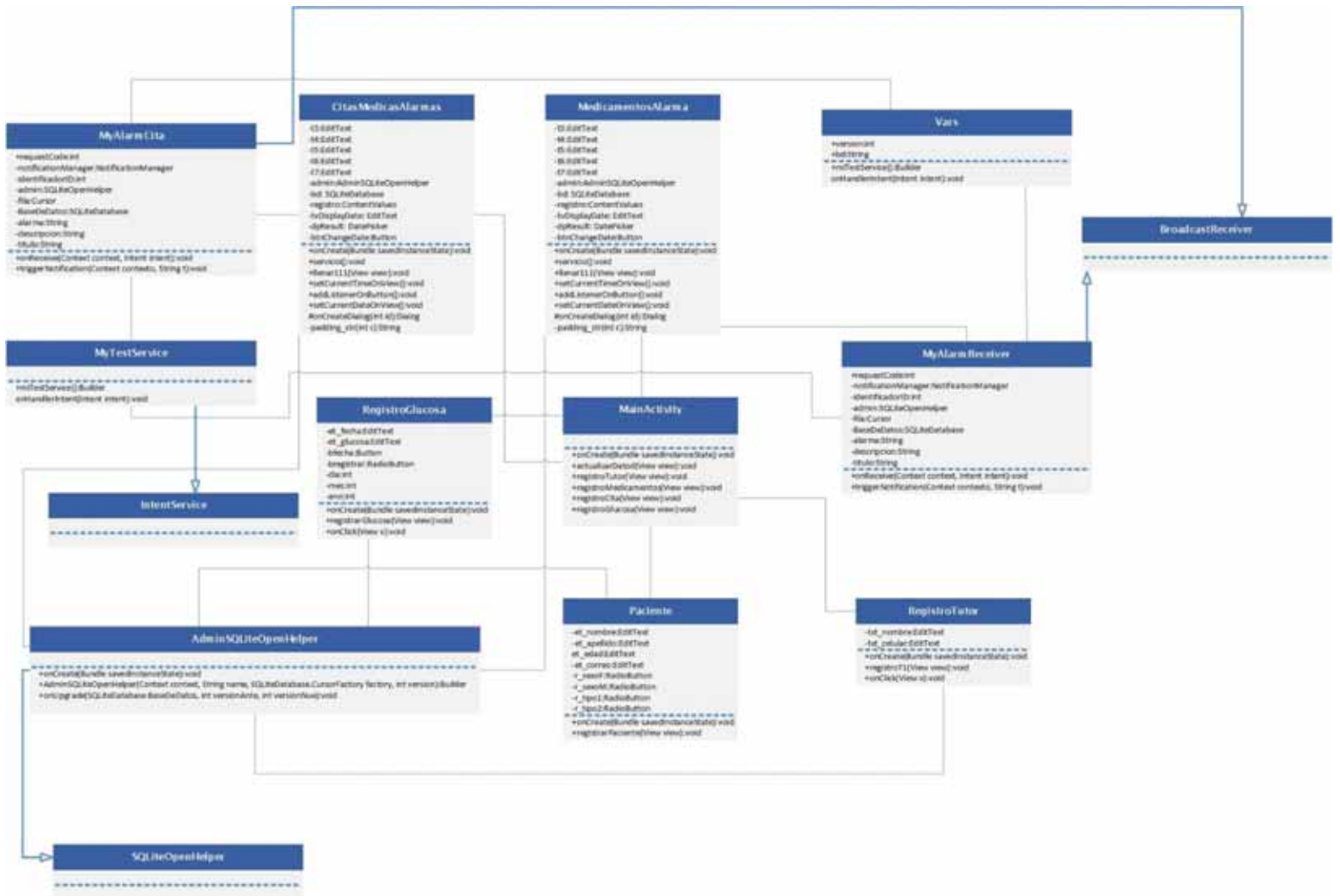


Figura 22. Diagrama UML

Resultados.

Se concluyó de manera satisfactoria la implementación de los siguientes módulos:

- Registro de paciente: El paciente puede registrar su nombre completo por medio de dos campos, su correo electrónico, su sexo y el tipo de diabetes que padece.
- Registro de medicamentos: Puede registrar el nombre del medicamento, su fecha de inicio, la hora y un mensaje por si gusta recordar la dosis correspondiente.
- Registro de tutor: Permite registrar a un tutor por medio de un nombre y un número de celular.
- Registro de citas médicas: Registra la cita, la especialidad, la fecha y la hora para un mejor control de la enfermedad con un especialista.
- Registro de glucosa: Ofrece registrar la medida de glucosa, así como la fecha que se tomó dicha medida.
- Alerta de medicamentos: El dispositivo sonará de acuerdo a la hora que el paciente haya registrado el medicamento.
- Alerta de cita médica: El dispositivo sonará de acuerdo a la hora y fecha de la cita médica.

Nota: Todos los campos están validados, en caso de estar algún campo vacío, no se registrará alguna solicitud.

No se concluyeron el módulo de alerta al tutor y la gráfica de glucosa por falta de tiempo en concretar el proyecto.

Conclusiones

La aplicación tiene la finalidad de mejorar la calidad de vida del paciente diabético utilizando la tecnología de hoy en día gracias a las aplicaciones móviles.

Se obtuvieron resultados positivos que permitirá al paciente llevar un mejor control de su enfermedad gracias a las alertas que emite la aplicación, probablemente disminuirá el número de ocasiones que por descuido olvide tomar su medicamento o alguna cita médica con el especialista.

El uso de la aplicación concientizará al paciente a tener un mejor cuidado de su enfermedad, así como un chequeo con un especialista.

No fue fácil implementar la aplicación, ya que en el transcurso de la carrera de computación no existió como tal alguna UEA que se dedicara al desarrollo de aplicaciones móviles, sin embargo existieron herramientas de diversas materias que ofrecieron las herramientas necesarias para concretar este proyecto.

Anexo A. Código fuente

Clase: AdminSQLiteOpenHelper

```

package com.example.usuario.diabetespt;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

//Clase administradora de la base de datos.
public class AdminSQLiteOpenHelper extends SQLiteOpenHelper{

    public AdminSQLiteOpenHelper(Context context, String name,
    SQLiteDatabase.CursorFactory factory, int version) { //método
    constructor

        super(context, name, factory, version);
    }

    //creación de la tabla de la base de datos
    @Override
    public void onCreate(SQLiteDatabase BaseDeDatos) {

        BaseDeDatos.execSQL("create table medicamentos(nombre text
primary key, cantidad int, dosis int)");
        BaseDeDatos.execSQL("create table tutores(celular text primary
key, nombre text)");
        BaseDeDatos.execSQL("create table paciente(nombre text primary
key, apellido text, edad int, correo text, sexo int, tipo int)");
        BaseDeDatos.execSQL("create table glucosa(medida text primary
key, fecha text)");
        BaseDeDatos.execSQL("create table citas(hora primary key,
fecha text)");
        BaseDeDatos.execSQL("create table alarma( idal integer primary
key autoincrement,encabezado text, mensaje text,fecha date, hora
time)");
        BaseDeDatos.execSQL("create table alarmaCita( idal integer
primary key autoincrement,encabezado text, mensaje text,fecha date,
hora time)");

    }

    @Override
    public void onUpgrade(SQLiteDatabase BaseDeDatos, int versionAnte,
int versionNue) {
        BaseDeDatos.execSQL("drop table if exists alarma" );
        BaseDeDatos.execSQL(" create table alarma( idal integer
primary key autoincrement,encabezado text, mensaje text,fecha date,
hora time)");
        BaseDeDatos.execSQL(" create table alarmaCita( idal integer
primary key autoincrement,encabezado text, mensaje text,fecha date,
hora time)");

    }
}

```

Clase: CitasMedicasAlarmas

```
//Clase para la gestión de las citas médicas.
package com.example.usuario.diabetespt;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import android.app.AlarmManager;
import android.app.DatePickerDialog;
import android.app.Dialog;
import android.app.PendingIntent;
import android.app.TimePickerDialog;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;

import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;

import java.util.Calendar;

public class CitasMedicasAlarmas extends AppCompatActivity {

    private EditText t4, t3,t5,t6,t7;
    private AdminSQLiteOpenHelper admin;
    private SQLiteDatabase bd;
    private ContentValues registro;
    private EditText tvDisplayDate;
    private DatePicker dpResult;
    private Button btnChangeDate;

    //fecha
    private int year;
    private int month;
    private int day;
    static final int DATE_DIALOG_ID = 999;
    // hora
    private int minute;
    private int hour;
    private TimePicker timePicker;
    private TextView textViewTime;
    private Button button;
    private static final int TIME_DIALOG_ID = 998;

    Calendar calendario = Calendar.getInstance();
    int hora, min,dia,mes,ano,horall;
    String cadenaF, cadenaH,fecha_sistema,hora_sistema;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_citas_medicas_alarmas);
    }
}
```

```

        admin = new AdminSQLiteOpenHelper(this, vars.bd, null,
vars.version);
        SQLiteDatabase BaseDeDatos = admin.getWritableDatabase();
        dia = calendario.get(Calendar.DAY_OF_MONTH);
        mes = calendario.get(Calendar.MONTH)+1;
        ano = calendario.get(Calendar.YEAR);
        hora = calendario.get(Calendar.HOUR_OF_DAY);
        min = calendario.get(Calendar.MINUTE);
        fecha_sistema=mes+"-"+dia+"-"+ano+" ";
        hora_sistema = hora+": "+min;

        setCurrentDateOnView();
        addListenerOnButton();
        // hora
        setCurrentTimeOnView();
        t3 = (EditText) findViewById(R.id.editText21);
        t5= (EditText) findViewById(R.id.editText22);
        t6= (EditText) findViewById(R.id.editText23);
        t7= (EditText) findViewById(R.id.editText24);
        servicio();
    }

    public void servicio() {
        Intent intent = new Intent(getApplicationContext(),
MyAlarmReceiverCita.class);
        final PendingIntent pIntent = PendingIntent.getBroadcast(this,
MyAlarmReceiverCita.REQUEST_CODE, intent,
PendingIntent.FLAG_UPDATE_CURRENT);
        long firstMillis = System.currentTimeMillis();
        int intervalMillis = 1 * 3 * 1000; //3 segundos
        AlarmManager alarm = (AlarmManager)
this.getSystemService(Context.ALARM_SERVICE);
        alarm.setInexactRepeating(AlarmManager.RTC_WAKEUP,
firstMillis, intervalMillis, pIntent);
    }

    public void llenar111(View view) {
        AdminSQLiteOpenHelper admin = new
AdminSQLiteOpenHelper(this,vars.bd, null, vars.version);
        SQLiteDatabase bd = admin.getReadableDatabase();
        bd = admin.getWritableDatabase();
        registro = new ContentValues();
        registro.put("encabezado", t3.getText().toString());
        registro.put("mensaje", t5.getText().toString()); //nombre del
campo
        registro.put("fecha", t6.getText().toString());
        registro.put("hora", t7.getText().toString());
        bd.insert("alarmaCita", null, registro); //nombre de la tabla
        bd.close();
        t3.setText("");
        t5.setText("");
        t6.setText("");
        t7.setText("");
        Toast.makeText(this, "alarma registrada",
Toast.LENGTH_LONG).show();
    }

    public void setCurrentTimeOnView() {

```

```

        textViewTime = (EditText) findViewById(R.id.editText24);

        timePicker = (TimePicker) findViewById(R.id.timePicker);
        final Calendar c = Calendar.getInstance();
        hour = c.get(Calendar.HOUR_OF_DAY);
        minute = c.get(Calendar.MINUTE);
        timePicker.setCurrentHour(hour);
        timePicker.setCurrentMinute(minute);

    }

    public void addListenerOnButton() {

        btnChangeDate = (Button) findViewById(R.id.button28);
        btnChangeDate.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                showDialog(DATE_DIALOG_ID);

            }

        });
        // hora

        button = (Button) findViewById(R.id.button29);
        button.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                showDialog(TIME_DIALOG_ID);

            }

        });

    }

    // hora

    public void setCurrentDateOnView() {

        tvDisplayDate = (EditText) findViewById(R.id.editText23);
        dpResult = (DatePicker) findViewById(R.id.datePicker);

        final Calendar c = Calendar.getInstance();
        year = c.get(Calendar.YEAR);
        month = c.get(Calendar.MONTH);
        day = c.get(Calendar.DAY_OF_MONTH);

        tvDisplayDate.setText(new StringBuilder()

            .append(month + 1).append("-").append(day).append("-")
            .append(year).append(" "));
    }

```

```

        dpResult.init(year, month, day, null);
    }

    @Override
    protected Dialog onCreateDialog(int id) {
        switch (id) {
            case DATE_DIALOG_ID:

                return new DatePickerDialog(this, datePickerListener,
                    year, month, day);
            case TIME_DIALOG_ID:

                return new TimePickerDialog(this, timePickerListener,
hour, minute, false);

        }
        return null;
    }

    // hora
    private TimePickerDialog.OnTimeSetListener timePickerListener =
new TimePickerDialog.OnTimeSetListener() {

        public void onTimeSet(TimePicker view, int selectedHour, int
selectedMinute) {

            hour = selectedHour;

            minute = selectedMinute;
            textViewTime.setText(new
StringBuilder().append(padding_str(hour)).append(":").append(padding_s
tr(minute)));
            timePicker.setCurrentHour(hour);
            timePicker.setCurrentMinute(minute);

        }

    };

    private static String padding_str(int c) {

        if (c >= 10)
            return String.valueOf(c);

        else

            return "0" + String.valueOf(c);

    }

    private DatePickerDialog.OnDateSetListener datePickerListener
= new DatePickerDialog.OnDateSetListener() {

        public void onDateSet(DatePicker view, int selectedYear,
            int selectedMonth, int selectedDay) {

```

```

        year = selectedYear;
        month = selectedMonth;
        day = selectedDay;

        tvDisplayDate.setText(new StringBuilder().append(month +
1)         .append("-").append(day).append("-").append(year)
         .append(" "));

        dpResult.init(year, month, day, null);
    }
};
}

```

Clase: MainActivity

```

//Clase que gestiona el menú principal que visualiza el usuario.
package com.example.usuario.diabetespt;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void actualizarDatos(View view){

        Intent actDatos = new Intent(this, Paciente.class);
        startActivity(actDatos);

    }

    public void registroTutor(View view){

        Intent regTutor = new Intent(this, RegistroTutor1.class);
        startActivity(regTutor);

    }

    public void registroMedicamento(View view){

        Intent regMed = new Intent(this, MedicamentosAlarma.class);
        startActivity(regMed);

    }

    public void registroCita(View view){

        Intent regCita = new Intent(this, CitasMedicasAlarmas.class);
        startActivity(regCita);

    }
}

```

```

public void registroGlucosa(View view){

    Intent regGluc = new Intent(this, RegistroGlucosa.class);
    startActivity(regGluc);

}

```

Clase: MedicamentosAlarma

//Clase que gestiona la hora y alarma para tomar el medicamento el //paciente.

```

package com.example.usuario.diabetespt;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import android.app.AlarmManager;
import android.app.DatePickerDialog;
import android.app.Dialog;
import android.app.PendingIntent;
import android.app.TimePickerDialog;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;

import java.util.Calendar;

public class MedicamentosAlarma extends AppCompatActivity {

    private EditText t4, t3,t5,t6,t7;
    private AdminSQLiteOpenHelper admin;
    private SQLiteDatabase bd;
    private ContentValues registro;
    private EditText tvDisplayDate;
    private DatePicker dpResult;
    private Button btnChangeDate;
    //fecha
    private int year;
    private int month;
    private int day;

    static final int DATE_DIALOG_ID = 999;
    // hora

```



```

private int minute;
private int hour;
private TimePicker timePicker;
private TextView textViewTime;
private Button button;
private static final int TIME_DIALOG_ID = 998;

Calendar calendario = Calendar.getInstance();
int hora, min, dia, mes, ano, horall;
String cadenaF, cadenaH, fecha_sistema, hora_sistema;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_medicamentos_alarma);
    admin = new AdminSQLiteOpenHelper(this, vars.bd, null,
vars.version);
    SQLiteDatabase BaseDeDatos = admin.getWritableDatabase();
    dia = calendario.get(Calendar.DAY_OF_MONTH);
    mes = calendario.get(Calendar.MONTH)+1;
    ano = calendario.get(Calendar.YEAR);
    hora = calendario.get(Calendar.HOUR_OF_DAY);
    min = calendario.get(Calendar.MINUTE);
    fecha_sistema=mes+"-"+dia+"-"+ano+" ";
    hora_sistema = hora+":"+min;

    setCurrentDateOnView();
    addListenerOnButton();
    // hora
    setCurrentTimeOnView();
    t3 = (EditText) findViewById(R.id.editText21);
    t5= (EditText) findViewById(R.id.editText22);
    t6= (EditText) findViewById(R.id.editText23);
    t7= (EditText) findViewById(R.id.editText24);
    servicio();

}

public void servicio() {
    Intent intent = new Intent(getApplicationContext(),
MyAlarmReceiver.class);
    final PendingIntent pIntent = PendingIntent.getBroadcast(this,
MyAlarmReceiver.REQUEST_CODE, intent,
PendingIntent.FLAG_UPDATE_CURRENT);
    long firstMillis = System.currentTimeMillis();
    int intervalMillis = 1 * 3 * 1000; //3 segundos
    AlarmManager alarm = (AlarmManager)
this.getSystemService(Context.ALARM_SERVICE);
    alarm.setInexactRepeating(AlarmManager.RTC_WAKEUP,
firstMillis, intervalMillis, pIntent);
}

public void llenar111(View view) {
    AdminSQLiteOpenHelper admin = new
AdminSQLiteOpenHelper(this,vars.bd, null, vars.version);
    SQLiteDatabase bd = admin.getReadableDatabase();
    bd = admin.getWritableDatabase();
    registro = new ContentValues();
    registro.put("encabezado", t3.getText().toString());
}

```

```

registro.put("mensaje", t5.getText().toString()); //nombre del
campo
registro.put("fecha", t6.getText().toString());
registro.put("hora", t7.getText().toString());
bd.insert("alarma", null, registro); //nombre de la tabla
bd.close();
t3.setText("");
t5.setText("");
t6.setText("");
t7.setText("");
Toast.makeText(this, "alarma registrada",
Toast.LENGTH_LONG).show();
}

public void setCurrentTimeOnView() {

    textViewTime = (EditText) findViewById(R.id.editText24);

    timePicker = (TimePicker) findViewById(R.id.timePicker);
    final Calendar c = Calendar.getInstance();
    hour = c.get(Calendar.HOUR_OF_DAY);
    minute = c.get(Calendar.MINUTE);
    timePicker.setCurrentHour(hour);
    timePicker.setCurrentMinute(minute);

}

public void addListenerOnButton() {

    btnChangeDate = (Button) findViewById(R.id.button28);
    btnChangeDate.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            showDialog(DATE_DIALOG_ID);

        }

    });
    // hora

    button = (Button) findViewById(R.id.button29);
    button.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v) {

            showDialog(TIME_DIALOG_ID);

        }

    });

}

// hora

public void setCurrentDateOnView() {

```

```

tvDisplayDate = (EditText) findViewById(R.id.editText23);
dpResult = (DatePicker) findViewById(R.id.datePicker);

final Calendar c = Calendar.getInstance();
year = c.get(Calendar.YEAR);
month = c.get(Calendar.MONTH);
day = c.get(Calendar.DAY_OF_MONTH);

tvDisplayDate.setText(new StringBuilder()

    .append(month + 1).append("-").append(day).append("-")
    .append(year).append(" "));

dpResult.init(year, month, day, null);

}

@Override
protected Dialog onCreateDialog(int id) {
    switch (id) {
        case DATE_DIALOG_ID:

            return new DatePickerDialog(this, datePickerListener,
                year, month, day);
        case TIME_DIALOG_ID:

            return new TimePickerDialog(this, timePickerListener,
hour, minute, false);

    }
    return null;
}

// hora
private TimePickerDialog.OnTimeSetListener timePickerListener =
new TimePickerDialog.OnTimeSetListener() {

    public void onTimeSet(TimePicker view, int selectedHour, int
selectedMinute) {

        hour = selectedHour;

        minute = selectedMinute;
        textViewTime.setText(new
StringBuilder().append(padding_str(hour)).append(":").append(padding_s
tr(minute)));
        timePicker.setCurrentHour(hour);
        timePicker.setCurrentMinute(minute);

    }

};

```

```

private static String padding_str(int c) {
    if (c >= 10)
        return String.valueOf(c);
    else
        return "0" + String.valueOf(c);
}

private DatePickerDialog.OnDateSetListener datePickerListener
    = new DatePickerDialog.OnDateSetListener() {

    public void onDateSet(DatePicker view, int selectedYear,
        int selectedMonth, int selectedDay) {
        year = selectedYear;
        month = selectedMonth;
        day = selectedDay;

        tvDisplayDate.setText(new StringBuilder().append(month +
1)         .append("-").append(day).append("-").append(year)
            .append(" "));

        dpResult.init(year, month, day, null);
    }
};
}

```

Clase: MyAlarmReceiver

//Clase que gestiona las notificaciones que se le muestra al paciente,
//al igual que como interactua con la base de datos (MEDICAMENTOS).

```

package com.example.usuario.diabetespt;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.BitmapFactory;
import android.media.RingtoneManager;
import android.net.Uri;
import android.support.v4.app.NotificationCompat;

import java.util.Calendar;

public class MyAlarmReceiver extends BroadcastReceiver {

```

```

public static final int REQUEST_CODE = 12345;
private NotificationManager notificationManager;
private final int NOTIFICATION_ID = 1010;
private AdminSQLiteOpenHelper admin;
private Cursor fila;
private SQLiteDatabase BaseDeDatos;
private String alarma,descripcion,titulo;

@Override
public void onReceive(Context context, Intent intent) {

    Intent i = new Intent(context, MyTestService.class);
    context.startService(i);
    Calendar calendario = Calendar.getInstance();
    int hora, min,dia,mes,ano;
    String cadenaF, cadenaH,fecha_sistema,hora_sistema;

    dia = calendario.get(Calendar.DAY_OF_MONTH);
    mes = calendario.get(Calendar.MONTH)+1;
    ano = calendario.get(Calendar.YEAR);
    hora = calendario.get(Calendar.HOUR_OF_DAY);
    min = calendario.get(Calendar.MINUTE);
    fecha_sistema=mes+"-"+dia+"-"+ano+" ";
    hora_sistema=hora+": "+min;
    admin = new AdminSQLiteOpenHelper(context, vars.bd, null,
vars.version);
    BaseDeDatos = admin.getWritableDatabase();

    if(BaseDeDatos!=null) {
        fila = BaseDeDatos.rawQuery("SELECT * FROM alarma WHERE
fecha='"+fecha_sistema+"' AND hora= '"+hora_sistema+"'", null);
        if(fila.moveToFirst()){
            alarma=fila.getString(0);
            titulo=fila.getString(1);
            descripcion =fila.getString(2);
            triggerNotification(context,titulo+"\n"+descripcion);
        }
    }
    BaseDeDatos.close();

}

private void triggerNotification(Context contexto, String t) {
    Intent notificationIntent = new Intent(contexto,
MainActivity.class);
    notificationIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    PendingIntent contentIntent =
PendingIntent.getActivity(contexto, 0, notificationIntent,
PendingIntent.FLAG_UPDATE_CURRENT);
    Uri defaultSound =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
    long[] pattern = new long[]{2000, 1000, 2000};

    NotificationCompat.Builder builder = new
NotificationCompat.Builder(contexto);
    builder.setContentIntent(contentIntent)

        .setTicker("")
        .setContentTitle("alarma ")
        .setContentTitle("")

```

```

        .setContentText(t)
        .setContentInfo("Info")

    .setLargeIcon(BitmapFactory.decodeResource(contexto.getResources(),
R.drawable.ic_notificacion))
        .setSmallIcon(R.drawable.ic_notificacion)
        .setAutoCancel(true) //Cuando se pulsa la notificación
ésta desaparece
        .setSound(defaultSound)
        .setVibrate(pattern);

    Notification notificacion = new
NotificationCompat.BigTextStyle(builder.setContentIntent(contentIntent
))
        .bigText(t)
        .setBigContentTitle("Medicamento")
        .setSummaryText("Alerta de Medicamento")
        .build();

    notificationManager = (NotificationManager)
contexto.getSystemService(Context.NOTIFICATION_SERVICE);
    notificationManager.notify(NOTIFICATION_ID, notificacion);
}
}

```

Clase: MyAlarmReceiverCita

//Clase que gestiona las notificaciones que se le muestra al paciente,
//al igual que como interactua con la base de datos.(CITAS MÉDICAS)

```

package com.example.usuario.diabetespt;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.BitmapFactory;
import android.media.RingtoneManager;
import android.net.Uri;
import android.support.v4.app.NotificationCompat;

import java.util.Calendar;

public class MyAlarmReceiverCita extends BroadcastReceiver {

    public static final int REQUEST_CODE = 12345;
    private NotificationManager notificationManager;
    private final int NOTIFICATION_ID = 1010;
    private AdminSQLiteOpenHelper admin;
    private Cursor fila;
    private SQLiteDatabase BaseDeDatos;
    private String alarma,descripcion,titulo;

```

```

@Override
public void onReceive(Context context, Intent intent) {

    Intent i = new Intent(context, MyTestService.class);
    context.startService(i);
    Calendar calendario = Calendar.getInstance();
    int hora, min, dia, mes, ano;
    String cadenaF, cadenaH, fecha_sistema, hora_sistema;

    dia = calendario.get(Calendar.DAY_OF_MONTH);
    mes = calendario.get(Calendar.MONTH)+1;
    ano = calendario.get(Calendar.YEAR);
    hora = calendario.get(Calendar.HOUR_OF_DAY);
    min = calendario.get(Calendar.MINUTE);
    fecha_sistema=mes+"-"+dia+"-"+ano+" ";
    hora_sistema=hora+": "+min;
    admin = new AdminSQLiteOpenHelper(context, vars.bd, null,
vars.version);
    BaseDeDatos = admin.getWritableDatabase();

    if(BaseDeDatos!=null) {
        fila = BaseDeDatos.rawQuery("SELECT * FROM alarmaCita
WHERE fecha='"+fecha_sistema+"' AND hora= '"+hora_sistema+"'", null);
        if(fila.moveToFirst()){
            alarma=fila.getString(0);
            titulo=fila.getString(1);
            descripcion =fila.getString(2);
            triggerNotification(context,titulo+"\n"+descripcion);
        }
    }
    BaseDeDatos.close();

}

private void triggerNotification(Context contexto, String t) {
    Intent notificationIntent = new Intent(contexto,
MainActivity.class);
    notificationIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    PendingIntent contentIntent =
PendingIntent.getActivity(contexto, 0, notificationIntent,
PendingIntent.FLAG_UPDATE_CURRENT);
    Uri defaultSound =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
    long[] pattern = new long[]{2000, 1000, 2000};

    NotificationCompat.Builder builder = new
NotificationCompat.Builder(contexto);
    builder.setContentIntent(contentIntent)

        .setTicker("")
        .setContentTitle("alarma ")
        .setContentTitle("")
        .setContentText(t)
        .setContentInfo("Info")

    .setLargeIcon(BitmapFactory.decodeResource(contexto.getResources(),
R.drawable.ic_notificacion))
        .setSmallIcon(R.drawable.ic_notificacion)
        .setAutoCancel(true) //Cuando se pulsa la notificación
esta desaparece
        .setSound(defaultSound)

```

```

        .setVibrate(pattern);

        Notification notificacion = new
NotificationCompat.BigTextStyle(builder.setContentIntent(contentIntent
))
        .bigText(t)
        .setBigContentTitle("Cita Médica")
        .setSummaryText("Alerta de Citas")
        .build();

        notificationManager = (NotificationManager)
contexto.getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify(NOTIFICATION_ID, notificacion);
    }
}

```

Clase: MyTestService

//Clase que proporciona el servicio para solicitudes asíncronas

```

package com.example.usuario.diabetespt;

import android.app.IntentService;
import android.content.Intent;
import android.util.Log;

public class MyTestService extends IntentService {

    public MyTestService() {
        super("MyTestService");
    }

    @Override
    protected void onHandleIntent(Intent intent) {
        Log.i("MyTestService", "Servicio ejecutandose.
Recordatorios");
    }
}

```

Clase: Paciente

//Clase que contiene todos los datos personales del paciente.

```

package com.example.usuario.diabetespt;

import android.content.ContentValues;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.Toast;

public class Paciente extends AppCompatActivity {

    private EditText et_nombre, et_apellido, et_edad, et_correo;
}

```



```

private RadioButton r_sexoM, r_sexoF, r_tipo1, r_tipo2;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_paciente);

    et_nombre = (EditText) findViewById(R.id.txt_nombre);
    et_apellido = (EditText) findViewById(R.id.txt_apellido);
    et_edad = (EditText) findViewById(R.id.txt_edadScroll);
    et_correo = (EditText) findViewById(R.id.txt_mailScroll);
    r_sexoM = (RadioButton) findViewById(R.id.rb_masculinoScroll);
    r_sexoF = (RadioButton) findViewById(R.id.rb_femeninoScroll);
    r_tipo1 = (RadioButton) findViewById(R.id.rb_tipo1Scroll);
    r_tipo2 = (RadioButton) findViewById(R.id.rb_tipo2Scroll);
}

public void registrarPaciente(View view){

    AdminSQLiteOpenHelper admin = new AdminSQLiteOpenHelper(this,
"diabetesBD", null, 1);

    //abrir la base de datos en modo lectura y escritura
    SQLiteDatabase BaseDeDatos = admin.getWritableDatabase();

    String nombre = et_nombre.getText().toString();
    String apellido = et_apellido.getText().toString();
    String edad = et_edad.getText().toString();
    String correo = et_correo.getText().toString();
    String sexoM = r_sexoM.getText().toString();
    String sexoF = r_sexoF.getText().toString();
    String tipo1 = r_tipo1.getText().toString();
    String tipo2 = r_tipo2.getText().toString();

    if(!nombre.isEmpty() && !apellido.isEmpty() &&
!edad.isEmpty() && !correo.isEmpty() && !sexoM.isEmpty() &&
!sexoF.isEmpty() && !tipo1.isEmpty() && !tipo2.isEmpty()){
        ContentValues registro = new ContentValues();
        registro.put("nombre", nombre);
        registro.put("apellido", apellido);
        registro.put("edad", edad);
        registro.put("correo", correo);
        registro.put("sexoM", sexoM);
        registro.put("sexoF", sexoF);
        registro.put("tipo1", tipo1);
        registro.put("tipo2", tipo2);

        BaseDeDatos.insert("paciente", null, registro);
        BaseDeDatos.close();
        // et_nombre.setText("");

        Toast.makeText(this, "Registro exitoso",
Toast.LENGTH_SHORT).show();

    }else{
        Toast.makeText(this, "Ningún campo debe ser
vacío", Toast.LENGTH_SHORT).show();
    }

    Intent registro = new Intent(this, MainActivity.class);

```

```

        startActivity(registro);
    }
}

```

Clase: RegistroGlucosa

```

//Clase que gestiona las medidas de la glucosa
package com.example.usuario.diabetespt;

import android.app.DatePickerDialog;
import android.content.ContentValues;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.Toast;

import java.util.Calendar;

public class RegistroGlucosa extends AppCompatActivity implements
View.OnClickListener {

    Button bfecha, bregistrar;
    EditText et_fecha, et_glucosa;
    private int dia, mes, ano;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registro_glucosa);

        bfecha = (Button) findViewById(R.id.bfecha);
        et_fecha = (EditText) findViewById(R.id.txt_fechaGlucosa);
        et_glucosa = (EditText) findViewById(R.id.txt_regisGluc);
        bregistrar = (Button) findViewById(R.id.bregistroGlucosa);

        bfecha.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {

        if(v==bfecha){
            final Calendar c = Calendar.getInstance();
            ano = c.get(Calendar.YEAR);
            mes = c.get(Calendar.MONTH);
            dia = c.get(Calendar.DAY_OF_MONTH);

            DatePickerDialog datePickerDialog = new
            DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() {
                @Override
                public void onDateSet(DatePicker view, int year, int
            monthOfYear, int dayOfMonth) {

```

```

        et_fecha.setText(dayOfMonth + "/" +
(monthOfYear+1) + "/" +year);
    }
    },ano,mes,dia);
    datePickerDialog.show();

}

}

public void registrarGlucosa(View view){

    AdminSQLiteOpenHelper admin = new AdminSQLiteOpenHelper(this,
"diabetesBD", null, 1);

    //abrir la Base de datos en modo lectura y escritura
    SQLiteDatabase BaseDeDatos = admin.getWritableDatabase();

    String glucosa = et_glucosa.getText().toString();
    String fecha = et_fecha.getText().toString();

    if(!glucosa.isEmpty() && !fecha.isEmpty()){

        ContentValues registro = new ContentValues();
        registro.put("glucosa", glucosa);
        registro.put("fecha", fecha);

        BaseDeDatos.insert("glucosa", null, registro);
        BaseDeDatos.close();
        Toast.makeText(this, "Registro Exitoso",
Toast.LENGTH_SHORT).show();

    }else{
        Toast.makeText(this, "No pueden existir campos vacíos",
Toast.LENGTH_SHORT).show();
    }

    Intent registro = new Intent(this, MainActivity.class);
    startActivity(registro);

}

}

```

Clase: RegistroTutor1

```

package com.example.usuario.diabetespt;
//Clase que gestiona el registro del tutor tomando en cuenta el nombre
//y celular del mismo.
import android.content.ContentValues;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

```

```

public class RegistroTutor1 extends AppCompatActivity {

    private EditText txt_nombre, txt_celular;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registro_tutor1);

        txt_nombre = (EditText)findViewById(R.id.txt_tutor1);
        txt_celular = (EditText)findViewById(R.id.txt_numTut1);
    }

    public void registroT1(View view){

        AdminSQLiteOpenHelper admin = new AdminSQLiteOpenHelper(this,
"diabetesBD", null,1);
        SQLiteDatabase BaseDeDatos = admin.getWritableDatabase();

        String nombreTutor = txt_nombre.getText().toString();
        String celular = txt_celular.getText().toString();

        if(!nombreTutor.isEmpty() && !celular.isEmpty()){

            ContentValues registroTutor = new ContentValues();

            registroTutor.put("nombreTutor", nombreTutor);
            registroTutor.put("celular", celular);

            //Inserción

            BaseDeDatos.insert("tutores", null, registroTutor);
            BaseDeDatos.close();

            txt_nombre.setText("");
            txt_celular.setText("");

            Toast.makeText(this, "Registro exitoso",
Toast.LENGTH_SHORT).show();

        }else {

            Toast.makeText(this, "Debe llenar todos los campos",
Toast.LENGTH_SHORT).show();
        }

    }

    public void menuP(View view){

        Intent mp = new Intent(this,MainActivity.class);
        startActivity(mp);

    }
}

```

```
}
```

Clase: Vars

```
//Clase que trabaja con la versión de la base de datos.  
package com.example.usuario.diabetespt;  
  
public class vars {  
    public static int version = 7;  
    public static String bd= "Taller";  
}
```

Bibliografía

- [1] B. A. Chávez Flores, V. Márquez Pérez. “Sistema de apoyo clínico para el tratamiento del paciente diabético”, proyecto terminal (Lic. En Ingeniería en Cómputo) Universidad Autónoma Metropolitana Azcapotzalco, “División de Ciencias Básicas e Ingeniería”, 2012
- [2] G. Monroy Rodríguez, “Reingeniería del Sistema de Consulta y Diagnóstico médico veterinario SICODIVE”, proyecto terminal (Lic. En Ingeniería en Cómputo) Universidad Autónoma Metropolitana Azcapotzalco, “División de Ciencias Básicas e Ingeniería”, 2012.
- [3] F.J. García Maldonado, “Prototipo de una aplicación móvil para la gestión de síntomas de un paciente”, proyecto terminal (Lic. En Ingeniería en Cómputo) Universidad Autónoma Metropolitana Azcapotzalco, “División de Ciencias Básicas e Ingeniería”, 2012.
- [4] JFcorp “Optimizador Control Diabético”, versión 1.1.1, “Google Play Store,” https://play.google.com/store/apps/details?id=com.appcorp.controldiabetico&hl=es_419, 1 de agosto de 2015.
- [5] W. Castrillon, “Diabetes, Control y Manejo”, versión 1.3, “Google Play Store,” <https://play.google.com/store/apps/details?id=com.wilsoncastrillon.diabetes>, 16 de noviembre de 2016.
- [6] G. Wordong, “BG Monitor”, versión 8.0.1, “Google Play Store,” <https://play.google.com/store/apps/details?id=com.wonggordon.bgmonitor&hl=es>, 22 de abril de 2017.