

Control de documento

Nombre del proyecto	Healthec
Cierre de iteración	18 28/04/2023
Generador por	Carlos Antonio Madrigal Trejo
Aprobado por	Carlos Antonio Madrigal Trejo
Alcance de la distribución del documento	Control interno para todo el proyecto.

Índice

Sobre este documento	3
Resumen de la Iteración	4
Identificación	4
Hitos especiales	4
Artefactos y evaluación	4
Riesgos y problemas (Riesgo con su identificador si sucedió o no sucedió)	5
Asignación de recursos	6
Anexos	7
Anexo A.	7
Anexo B.	11
Anexo D.	19
Anexo E.	21
Anexo F.	23
Glosario de términos	26

Sobre este documento

La calidad se logra por medio de la revisión constante de las actividades que conducen desde la idea al producto. Al momento del cierre de una iteración es buen momento para hacer un alto, y evaluar lo logrado, los problemas encontrados y los retos a enfrentar.

El presente documento marca el final de la iteración I8, y contiene una evaluación de los artefactos y actividades realizadas durante la misma.

Se recogen también las impresiones y observaciones hechas durante el desarrollo de la iteración, así como el esfuerzo invertido en cada una de las disciplinas involucradas.

Resumen de la Iteración

Identificación

Código de la iteración	Fase a la que pertenece	Fecha de inicio	Fecha de cierre	Comentarios
I8	Implementación	24 /04/23	28/04/23	Se continuó con el desarrollo de la aplicación, las actividades faltantes se completarán en el siguiente sprint.

Hitos especiales

En este sprint se siguió el desarrollo de la aplicación de todos los módulos, tanto en el front end con el diseño de la aplicación como en el back-end con la codificación lógica de la aplicación, se lograron avances sin embargo quedarán pendientes actividades para el siguiente sprint.

Artefactos y evaluación

Artefacto	Meta (%)	Comentarios
PROG-04	85%	Se consiguió desarrollar un poco más este módulo, se seguirá el desarrollo en el siguiente sprint.
PROG-05	75%	Se desarrolló un poco más el módulo, sin embargo aún faltan puntos que se desarrollaran en el siguiente sprint.
PROG-06	70%	Se desarrolló un poco más el módulo, sin embargo aún faltan puntos que se desarrollaran en el siguiente sprint.
PROG-07	85%	Se desarrolló la implementación gráfica del módulo y su funcionalidad, la conexión con la base de datos se continuará en el siguiente sprint.
PROG-08	25%	Se desarrolló la implementación gráfica del módulo, se seguirá el desarrollo en el siguiente sprint.

Artefacto	Aspecto a evaluar	Evaluación	Comentarios
PROG-04	85%	Programación	Se consiguió desarrollar un poco más este módulo, se seguirá el desarrollo en el siguiente sprint.
PROG-05	75%	Programación	Se desarrolló un poco más el módulo, sin embargo aún faltan puntos que se desarrollaran en el siguiente sprint.
PROG-05	70%	Programación	Se desarrolló un poco más el módulo, sin embargo aún faltan puntos que se desarrollaran en el siguiente sprint.
PROG-06	85%	Programación	Se desarrolló la implementación para desplegar la información, en el siguiente sprint se concluirá el módulo integrando la información desde la base de datos.
PROG-07	25%	Programación	Se desarrolló la implementación gráfica del módulo, se seguirá el desarrollo en el siguiente sprint.

Riesgos y problemas (Riesgo con su identificador si sucedió o no sucedió)

Notas y observaciones

	Riesgo	Ocurrió
RIE-02	Fallas de hardware	No
RIE-04	Enfermedades	No
RIE-10	Falta de ética y moral del personal	No

RIE-12	Requisitos confusos o ambiguos	No
RIE-14	Ambiente laboral deficiente	No
RIE-16	Documentación deficiente	No
RIE-19	Estimación del tiempo inadecuada	No
RIE-21	Falta de comunicación con el cliente	No
RIE-22	Falta de claridad en los roles de actividades	No
RIE-23	Falta de experiencia del líder de proyecto	No

Asignación de recursos

Rol	Horas-Hombre	Desempeñado por(área)	Observaciones
Programador	2 horas.	Carlos Antonio Madrigal Trejo	Se desempeñó de manera correcta.
Programador	2 horas.	Fernando Pérez Romero	Realizó un buen trabajo.
Programador	3 horas.	Omar Adrián Tapia Guzmán	Se esforzó por realizar sus tareas y avanzó satisfactoriamente.
Programador	2 horas.	Angel Dario Vidaña Vargas	Realizó un buen desarrollo .
Programador	2 horas.	Carlos Daniel López Romo	Hizo un buen desarrollo de sus actividades.

Anexos

Anexo A.

PROG-04 Programar activity de Meditación para dormir

AcvitivityMeditación.java

```
package com.example.appdefinicionesoftware;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.SearchView;
import android.widget.TextView;
import android.widget.Toast;

public class MeditacionActivity extends AppCompatActivity {

    ListView lv_definicionesList;
    SearchView sv_buscar;
    DataBase db;
    ArrayAdapter definicionesArrayAdapter;

    ListaAdaptador adaptador;
    ArrayAdapter adapter;

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

        lv_definicionesList = findViewById(R.id.lstv_definiciones);
        sv_buscar = findViewById(R.id.sv_buscar);

        db = new DataBase (MainActivity.this);

        adaptador = new ListaAdaptador(MainActivity.this, db);

        lv_definicionesList.setAdapter(adaptador);

        //-----
        lv_definicionesList.setOnItemClickListener(new AdapterView.OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

                LinearLayout layout = (LinearLayout)
                MainActivity.this.getLayoutInflater().inflate(R.layout.mostrar_layout, null );

                AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);

                TextView txtNombre = layout.findViewById(R.id.txv_mostrarNombre);
                TextView txtSiglas = layout.findViewById(R.id.txv_mostrarSiglas);
                TextView txtDescripcion = layout.findViewById(R.id.txv_mostrarDescripcion);

                DefinicionesModel def = (DefinicionesModel)
                parent.getItemAtPosition(position);

                txtNombre.setText(def.getNombreDefinicion());
                txtSiglas.setText(def.getSiglasDefinicion());
                txtDescripcion.setText(def.getDescripcionDefinicion());
            }
        });
    }
}
```

```

        builder.setView(layout).setPositiveButton("Regresar", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        dialogInterface.dismiss();
    }
}).setNegativeButton("Editar", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        LinearLayout layout = (LinearLayout)
MainActivity.this.getLayoutInflater().inflate(R.layout.add_edit_layout, null );

        EditText edtNombre =
layout.findViewById(R.id.edt_nombreDefinición);
        EditText edtSiglas =
layout.findViewById(R.id.edt_siglasDefinicion);
        EditText edtDescripcion =
layout.findViewById(R.id.edt_descripciónDefinición);

        edtNombre.setText(def.getNombreDefinicion());
        edtSiglas.setText(def.getSiglasDefinicion());
        edtSiglas.setText(def.getDescripcionDefinicion());

        AlertDialog.Builder builder = new
AlertDialog.Builder(MainActivity.this);

        builder.setView(layout).setPositiveButton("Ok", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                DefinicionesModel definicion;

                String nombre = edtNombre.getText().toString();
                String siglas = edtSiglas.getText().toString();
                String descripcion = edtDescripcion.getText().toString();

                try{
                    definicion = new DefinicionesModel(-1, nombre, siglas,
descripcion);
                } catch (Exception e){
                    definicion = new DefinicionesModel(-1, "error",
"error", "error");
                    Toast.makeText(MainActivity.this, "Es necesario
rellenar todos los campos", Toast.LENGTH_SHORT).show();
                }

                if(definicion.getSiglasDefinicion().equals("error")){
                    dialogInterface.cancel();
                }

                db.editOne(definicion);
                lv_definicionesList.setAdapter(adaptador);
            }
        }).setNegativeButton("Cancelar", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                Toast.makeText(MainActivity.this, "Cancelación de editar",
Toast.LENGTH_SHORT).show();
            }
        }).create().show();

    }
}).create().show();

});

//-----
-
lv_definicionesList.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {

```



```

@Override
    public boolean onItemClick(AdapterView<?> parent, View view, int position,
long id) {
        AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
        builder.setTitle("¿Deseas eliminar la
definición?").setPositiveButton("Aceptar", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                DefinicionesModel def = (DefinicionesModel)
parent.getItemAtPosition(position);
                db.deleteOne(def);
                lv_definicionesList.setAdapter(adaptador);
            }
        }).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                dialogInterface.dismiss();
            }
        }).create().show();
        return false;
    }
});

//-----
-
    sv_buscar.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String query) {
            adaptador.getFilter().filter(query);
            return false;
        }

        @Override
        public boolean onQueryTextChange(String query) {
            adaptador.getFilter().filter(query);
            return false;
        }
    });

}

//-----
-----
    public void btn_agregar (View v){
        LinearLayout layout = (LinearLayout)
MainActivity.this.getLayoutInflater().inflate(R.layout.add_edit_layout, null );

        EditText edtNombre = layout.findViewById(R.id.edt_nombreDefinición);
        EditText edtSiglas = layout.findViewById(R.id.edt_siglasDefinicion);
        EditText edtDescripcion = layout.findViewById(R.id.edt_descripciónDefinición);

        AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);

        builder.setView(layout).setPositiveButton("Ok", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                DefinicionesModel definicion;

                String nombre = edtNombre.getText().toString();
                String siglas = edtSiglas.getText().toString();
                String descripcion = edtDescripcion.getText().toString();

                try{
                    definicion = new DefinicionesModel(-1, nombre, siglas, descripcion);
                } catch (Exception e){
                    definicion = new DefinicionesModel(-1, "error", "error", "error");
                    Toast.makeText(MainActivity.this, "Es necesario rellenar todos los
campos", Toast.LENGTH_SHORT).show();
                }

                if(definicion.getSiglasDefinicion().equals("error")){
                    dialogInterface.cancel();
                }
            }
        });
    }
}

```

```
        db.addOne(definicion);
        lv_definicionesList.setAdapter(adaptador);

    }
    }).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            Toast.makeText(MainActivity.this, "Cancelación de registro",
Toast.LENGTH_SHORT).show();
        }
    }).create().show();
}
//-----
-----
}
```

Anexo B.

PROG-05 Programar activity de Recetas

ActivityRecetas.java

```
package com.example.appdefinicionessoftware;

import android.app.Activity;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.TextView;

import org.w3c.dom.Text;

import java.util.ArrayList;
import java.util.List;

public class ActivityRecetas extends BaseAdapter implements Filterable {

    Activity mActivity;
    DataBase db;

    public ListaAdaptador(Activity mActivity, DataBase db) {
        this.mActivity = mActivity;
        this.db = db;
    }

    @Override
    public int getCount() {
        return db.getDefiniciones().size();
    }

    @Override
    public DefinicionesModel getItem(int position) {
        return db.getDefiniciones().get(position);
    }

    @Override
    public long getItemId(int i) {
        return 0;
    }

    @Override
    public View getView(int position, View view, ViewGroup parent) {
        View definicionesLinea;
        LayoutInflater inflater =
            (LayoutInflater)mActivity.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        definicionesLinea = inflater.inflate(R.layout.definiciones_una_linea, parent,
            false);

        TextView tv_nombreDefinicion =
            definicionesLinea.findViewById(R.id.tv_NombreDefinicion);
        TextView tv_siglasDefinicion =
            definicionesLinea.findViewById(R.id.tv_SiglasDefinicion);

        DefinicionesModel def = this.getItem(position);
        tv_nombreDefinicion.setText(def.getNombreDefinicion());
        tv_siglasDefinicion.setText(def.getSiglasDefinicion());

        return definicionesLinea;
    }

    public Filter getFilter() {
        Filter filter = new Filter() {
            @Override
            protected FilterResults performFiltering(CharSequence constraint) {
```

```

FilterResults filterResults = new FilterResults();
if(constraint == null || constraint.length() == 0){
    filterResults.count = db.getDefiniciones().size();
    filterResults.values = db.getDefiniciones();
}
else{
    List<DefinicionesModel> resultsModel = new ArrayList<>();
    String searchStr = constraint.toString().toLowerCase();

    for(DefinicionesModel itemsModel:db.getDefiniciones()){
        if(itemsModel.getNombreDefinicion().contains(searchStr)){
            resultsModel.add(itemsModel);
        }
        filterResults.count = resultsModel.size();
        filterResults.values = resultsModel;
    }

    }

    return filterResults;
}

@Override
protected void publishResults(CharSequence constraint, FilterResults results) {

    //db.getDefiniciones() = (List<DefinicionesModel>) results.values;
    notifyDataSetChanged();

    }
};
return filter;
}
}

```

Anexo C.

PROG-06 Programar activity de Recordatorios médicos

activity_recordatorios.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/linearLayout2"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#FFF6E8"
android:clipChildren="true"
android:gravity="center|center_horizontal"
android:orientation="vertical"
tools:context=".RecordatoriosActivity">

    <Button
        android:id="@+id/button3"
        android:layout_width="160dp"
        android:layout_height="80dp"
        android:layout_marginTop="48dp"
        android:background="#F6DDB9"
        android:text="MI INFORMACION"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView2"
        app:layout_constraintVertical_bias="0.0" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="RECORDATORIO MEDICO"
        android:textSize="24sp"
        app:layout_constraintBottom_toTopOf="@+id/btnMedicamentos"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.72"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.103" />

    <ImageButton
        android:id="@+id/imageButton5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/textView2"
        app:layout_constraintHorizontal_bias="0.34"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.005"
        app:srcCompat="@android:drawable/ic_menu_revert" />

    <Button
        android:id="@+id/button4"
        android:layout_width="160dp"
        android:layout_height="80dp"
        android:background="#F6DDB9"
        android:text="CITAS"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.179"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.288" />

    <Button
        android:id="@+id/btnMedicamentos"
        android:layout_width="160dp"
```

```

        android:layout_height="80dp"
        android:background="#F6DDB9"
        android:text="MEDICAMENTOS"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.892"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.287" />

<TableLayout
    android:id="@+id/tableLayout"
    android:layout_width="374dp"
    android:layout_height="385dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.486"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.884">

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center">

        <TextView
            android:id="@+id/textView11"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="CITAS DE LA SEMANA"
            android:textSize="16sp" />
    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center">

        <EditText
            android:id="@+id/editTextTextPersonName4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10"
            android:gravity="center"
            android:inputType="textPersonName"
            android:text="Cita" />

        <CheckBox
            android:id="@+id/checkbox1"
            android:layout_width="62dp"
            android:layout_height="50dp" />
    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center">

        <EditText
            android:id="@+id/editTextTextPersonName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10"
            android:gravity="center"
            android:inputType="textPersonName"
            android:text="Cita" />

        <CheckBox
            android:id="@+id/checkbox10"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </TableRow>

```

```

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">

    <EditText
        android:id="@+id/editTextTextPersonName33"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:gravity="center"
        android:inputType="textPersonName"
        android:text="Cita" />

    <CheckBox
        android:id="@+id/checkboxMed4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">

    <TextView
        android:id="@+id/textView13"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="NOMBRE DEL MEDICAMENTO"
        android:textSize="16sp" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">

    <EditText
        android:id="@+id/editTextTextPersonName34"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:gravity="center"
        android:inputType="textPersonName"
        android:text="Medicamento" />

    <CheckBox
        android:id="@+id/checkboxMed3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">

    <EditText
        android:id="@+id/editTextTextPersonName35"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:gravity="center"
        android:inputType="textPersonName"
        android:text="Medicamento" />

    <CheckBox
        android:id="@+id/checkboxMed2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</TableRow>

<TableRow

```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center">

        <EditText
            android:id="@+id/editTextTextPersonName36"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10"
            android:gravity="center"
            android:inputType="textPersonName"
            android:text="Medicamento" />

        <CheckBox
            android:id="@+id/checkBox111"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center">

        <EditText
            android:id="@+id/editTextTextMed1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10"
            android:gravity="center"
            android:inputType="textPersonName"
            android:text="Medicamento" />

        <CheckBox
            android:id="@+id/checkBoxMed1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </TableRow>

</TableLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

RecordatoriosActivity.java

```

package mx.GPS.healthec;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.TableLayout;
import android.widget.TextView;
import android.widget.Toast;

public class RecordatoriosActivity extends AppCompatActivity {
    private Button miInfoBtn, citasBtn, medicamentosBtn;
    private ImageButton menuBtn;
    private TextView titleTxt;
    private TableLayout citasTable;

    private static final String[] MEDICAMENTOS = {"Ibuprofeno", "Paracetamol", "Aspirina",
    "Amoxicilina", "Omeprazol", "Diazepam"};

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

```



```
// Enlazar vistas con variables
miInfoBtn = findViewById(R.id.button3);
citasBtn = findViewById(R.id.button4);
medicamentosBtn = findViewById(R.id.btnMedicamentos);
menuBtn = findViewById(R.id.imageButton5);
titleTxt = findViewById(R.id.textView2);
citasTable = findViewById(R.id.tableLayout);


// Definir listener para el botón "Mi Información"
miInfoBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Aquí puedes agregar el código que se ejecutará al hacer clic en el botón
        Toast.makeText(RecordatoriosActivity.this, "Presionaste el botón Mi
Información", Toast.LENGTH_SHORT).show();
    }
});

// Definir listener para el botón "Citas"
citasBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Aquí puedes agregar el código que se ejecutará al hacer clic en el botón
        Toast.makeText(RecordatoriosActivity.this, "Presionaste el botón Citas",
Toast.LENGTH_SHORT).show();
    }
});

// Definir listener para el botón "Medicamentos"
Button btnMedicamentos = findViewById(R.id.btnMedicamentos);

btnMedicamentos.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        AlertDialog.Builder builder = new
AlertDialog.Builder(RecordatoriosActivity.this);
        builder.setTitle("Medicamentos")
            .setItems(MEDICAMENTOS, new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                    // Aquí puedes agregar la lógica para manejar la selección
                    del usuario
                }
            });
        AlertDialog dialog = builder.create();
        dialog.show();
    }
});

// Definir listener para el botón del menú
menuBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Aquí se agregara el código que se ejecutará al hacer clic en el botón del
        menú
        Toast.makeText(RecordatoriosActivity.this, "Presionaste el botón del menú",
Toast.LENGTH_SHORT).show();
    }
});
}
```

 RECORDATORIO MEDICO

MI INFORMACION

CITASMEDICAMENTOS

CITAS DE LA SEMANA

Cita

Cita

Cita

☐

☐

☐

NOMBRE DEL MEDICAMENTO

Medicamento

Medicamento

Medicamento

Medicamento

☐

☐

☐

☐

Anexo D.

PROG-07 Programar activity de Consejos saludables

ConsejosSaludablesActivity.java

```
package mx.GPS.healthec;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.TextView;

public class ConsejosSaludables extends AppCompatActivity {

    TextView txtvConsejo;

    ImageView imgvConsejo;

    ProgressBar prgsbImagenes;

    static int idSwitch=1;
    static int progress = 0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_consejos_saludables);

        txtvConsejo = (TextView)findViewById( R.id.txtvConsejo );
        imgvConsejo = (ImageView) findViewById(R.id.imgvConsejo);
        prgsbImagenes = findViewById(R.id.prgsbImagenes);
    }

    public void btnSiguienteClick (View v){

        progress+=10;
        idSwitch++;

        txtvConsejo.setText("Hola el consejo de hoy es comer frutas y verduras te hace una
persona mas saludable");

        prgsbImagenes.setProgress(progress,true);

        if(progress==100){
            progress=0;
            idSwitch = 1;
            prgsbImagenes.setProgress(progress,true);
        }
        switch (idSwitch){

            case 1: imgvConsejo.setImageResource(R.drawable.saludconsejo);
                    break;
            case 2: imgvConsejo.setImageResource(R.drawable.saludconsejo2);
                    break;
            case 3: imgvConsejo.setImageResource(R.drawable.saludconsejo3);
                    break;
            case 4: imgvConsejo.setImageResource(R.drawable.saludconsejo4);
                    break;
            case 5: imgvConsejo.setImageResource(R.drawable.saludconsejo5);
                    break;
            case 6: imgvConsejo.setImageResource(R.drawable.saludconsejo6);
                    break;
            case 7: imgvConsejo.setImageResource(R.drawable.saludconsejo7);
                    break;
            case 8: imgvConsejo.setImageResource(R.drawable.saludconsejo8);
                    break;
            case 9: imgvConsejo.setImageResource(R.drawable.saludconsejo9);
                    break;
        }
    }
}
```

```
case 10: imgvConsejo.setImageResource(R.drawable.saludconsejo10);  
        break;  
case 11: imgvConsejo.setImageResource(R.drawable.saludconsejo11);  
        break;  
  
    }  
  
    }  
  
}
```



Consejo Saludable

SIGUIENTE

Anexo E.

PROG-08 Programar activity de Información de sueño diario

activity_sleep.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFF6E8"
    android:gravity="center"
    android:orientation="vertical"
    tools:context=".SleepActivity">

    <TextView
        android:id="@+id/textView13"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif-black"
        android:gravity="center|center_vertical"
        android:padding="5dp"
        android:text="Información de tu sueño"
        android:textColor="#7A6969"
        android:textSize="24sp" />

    <EditText
        android:id="@+id/editTextTime"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dp"
        android:ems="10"
        android:inputType="time"
        android:singleLine="false"
        android:text="18:55"
        android:textAlignment="center"
        android:textColor="#7A6969"
        android:textSize="48sp" />

    <ToggleButton
        android:id="@+id/toggleButton"
        android:layout_width="223dp"
        android:layout_height="wrap_content"
        android:background="#F6DDB9"
        android:text="ToggleButton"
        android:textColor="#7A6969"
        android:textOff="Apagado"
        android:textOn="Encendido" />

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="346dp"
        android:layout_marginTop="25dp"
        android:layout_marginBottom="25dp">

        <ImageButton
            android:id="@+id/imageButton5"
            android:layout_width="147dp"
            android:layout_height="134dp"
            android:layout_alignParentStart="true"
            android:layout_marginStart="34dp"
            android:layout_marginEnd="56dp"
            android:layout_toStartOf="@+id/imageButton6"
            android:background="@null"
            android:scaleType="fitXY"
            android:src="@drawable/horario_sleep" />

        <ImageButton
            android:id="@+id/imageButton6"
            android:layout_width="140dp"
```

```

        android:layout_height="135dp"
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="34dp"
        android:background="@null"
        android:scaleType="fitXY"
        tools:srcCompat="@drawable/informacion_sleep" />

<ImageButton
    android:id="@+id/imageButton7"
    android:layout_width="wrap_content"
    android:layout_height="141dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="130dp"
    android:layout_marginEnd="129dp"
    android:layout_marginBottom="40dp"
    android:background="@null"
    android:scaleType="fitXY"
    tools:srcCompat="@drawable/configuracion_sleep" />
</RelativeLayout>

<Button
    android:id="@+id/button7"
    android:layout_width="223dp"
    android:layout_height="wrap_content"
    android:background="#F6DDB9"
    android:text="Comenzar a dormir" />
</LinearLayout>

```



Anexo F.

Tabla de McCall

CAPACIDAD	FACTOR	Métrica	Calificación
Operación	Corrección: Grado de cumplimiento de las especificaciones y objetivos del usuario	Compleción: Grado en que se logró implementar en la app los requerimientos especificados por las necesidades del usuario final	3
		Consistencia: Los requerimientos del usuario final cumplen con las técnicas de documentación por norma	3
		Trazabilidad: Los elementos del diseño de la app se identifican a partir de los requerimientos del usuario final	4
	Confiabilidad: Grado en el sistema está disponible para usarse.	Complejidad: La complejidad de las funciones de la app interfieren con la disponibilidad para usarse	4
		Consistencia: El diseño de la app permite el usuario utilizarla en cualquier momento	5
		Tolerancia a errores: Las fallas en la disponibilidad de la app interfieren en las necesidades de los usuarios	3
	Usabilidad: Grado de esfuerzo necesario que se requiere para aprender a utilizarlo.	Facilidad de formación: Facilidad en que el usuario puede aprender a utilizar la app	5
		Operatividad: La app cuenta con una guía para su facilidad de operación	3
	Integridad o Seguridad: Grado en el que se controla el acceso al programa o los datos por usuarios no autorizados.	Facilidad de auditoría: La app cuenta con las autenticaciones necesarias para impedir a usuarios no autorizados acceder a información sensible	4
		Instrumentación: La app cuenta con herramientas para identificar automáticamente brechas de seguridad en la información del usuario	5
		Seguridad: La app cuenta con elementos de protección para asegurar la protección de información importante.	3
	Eficiencia o Performance: Cantidad de recursos y código requeridos por un programa para	Concisión: Nivel de optimización de código en las funciones de la app.	3
		Eficiencia de ejecución: Nivel de eficiencia de ejecución en las funciones de la app.	4

	realizar una función.	Operatividad: Nivel de facilidad de operación en las funciones de la app.	3
Transición	Portabilidad: Grado que mide el esfuerzo para migrar un programa de un entorno de operación a otro.	Auto documentación: El código de la app cuenta con la claridad necesaria para portar en otro entorno sin documentación.	5
		Generalidad: La app en general es capaz de ser migrada a otro entorno.	2
		Modularidad: El código de la app es capaz de ser migrado por módulos.	2
	Reusabilidad: Grado de esfuerzo requerido para que el programa o una de sus partes pueda ser utilizado en otro proyecto.	Autodocumentación: El código de la app cuenta con la claridad necesaria para que otro desarrollador pueda utilizarlo en otro proyecto.	4
		Independencia hardware: La app o cualquier parte de su código puede ser utilizado en cualquier modelo celular.	4
		Independencia del sistema: La app o cualquier parte de su código puede ser utilizado en cualquier sistema operativo o versión vigente.	3
	Interoperabilidad: Grado de esfuerzo dedicado para que un sistema o programa pueda operar conjuntamente con otro.	Estd. Comunicaciones: Grado de uso de estándares para que la app pueda operar con otro software conjuntamente.	5
		Estandarización de datos: Nivel de manejo de interoperabilidad con otros softwares.	5
Revisión	Facilidad Mantenimiento: Esfuerzo requerido para localizar y corregir un error en un programa en funcionamiento.	Consistencia: Nivel de documentación empleada para reducir el esfuerzo requerido en corrección de errores.	4
		Modularidad: Nivel de modularidad de las funcionalidades para su fácil mantenimiento.	2
		Simplicidad: Nivel de simplicidad del código para su fácil mantenimiento por cualquier desarrollador	4

		incluso ajeno al proyecto.	
	Flexibilidad: Esfuerzo requerido para modificar un software en funcionamiento.	Capacidad de expansión: Grado permitido para ampliar la app en funcionamiento.	3
		Complejidad: Nivel de complejidad para ampliar la app en funcionamiento.	2
		Consistencia: Nivel de documentación empleada para poder ampliar la app en funcionamiento.	5
	Facilidad de Prueba: Grado de esfuerzo requerido para probar un programa verificando que realice adecuadamente sus funciones.	Auto documentación: La app puede ser probada debido a la claridad del código proporcionada por la documentación realizada.	3
		Facilidad de auditoría: Nivel de facilidad de auditoría de las funciones de la app.	5
		Instrumentación: Nivel de aplicación de herramientas que apliquen pruebas automatizadas a la app.	3

Glosario de términos

Hardware: Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.

Software: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

Interfaz: En el contexto de la programación, una interfaz es un conjunto de métodos y propiedades que se pueden implementar en una clase para permitir la comunicación con otras clases y componentes. Una interfaz define un contrato que debe cumplirse para que las clases puedan comunicarse entre sí de manera efectiva.

Hash: En el contexto de la seguridad informática, un hash es un valor numérico que se genera a partir de un conjunto de datos utilizando un algoritmo de hash. Los algoritmos de hash son funciones criptográficas que toman una entrada de datos y producen una salida de longitud fija que es única para esa entrada. Los hashes se utilizan para verificar la integridad de los datos, proteger las contraseñas y garantizar la autenticidad de los mensajes.

Salting: El salting (en español, "salteado") es una técnica de seguridad utilizada en la criptografía de contraseñas. Consiste en agregar una cadena aleatoria de datos a la contraseña antes de realizar el hash. El salting hace que las contraseñas sean más difíciles de romper mediante ataques de fuerza bruta o de diccionario, ya que el hash resultante será diferente para cada usuario. El proceso de salting y hashing juntos se conoce como "salting and hashing" o "hashing with salt".

Métodos: En programación, los métodos son bloques de código que realizan una tarea específica. Los métodos pueden recibir parámetros y devolver valores, lo que les permite interactuar con otros componentes del programa. Los métodos se utilizan para organizar y modularizar el código, lo que facilita el mantenimiento y la reutilización del mismo.

Base de datos: Una base de datos es un sistema de almacenamiento y recuperación de información. Las bases de datos se utilizan comúnmente en aplicaciones informáticas para almacenar información estructurada y permitir la recuperación y manipulación de los datos de manera eficiente.

Sensor: Un sensor es un componente de hardware que detecta y responde a cambios en su entorno. En el contexto de la programación móvil, los sensores se utilizan comúnmente en los dispositivos móviles para detectar la ubicación, el movimiento, la luz, la proximidad y otros aspectos del entorno del dispositivo. Los datos del sensor se pueden utilizar para crear experiencias interactivas en tiempo real y para mejorar la precisión de las aplicaciones.

Tareas en segundo plano: Las tareas en segundo plano son operaciones tipo "desencadenar y olvidar" y su progreso de ejecución no tiene ningún impacto en la interfaz de usuario o el proceso de llamada. Esto significa que el proceso de llamada no espera a la finalización de las tareas.