

Control de documento

Nombre del proyecto	Healthec
Cierre de iteración	19 05/05/2023
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)	6
Asignación de recursos	6
Anexos	7
Anexo A.	7
Anexo B.	13
Anexo C.	16
Anexo D.	23
Anexo E.	26
Anexo F.	28
Anexo G.	30
Glosario de términos	33

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 I9, 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
I9	Implementación	1 /05/23	05/05/23	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	90%	Se consiguió desarrollar un poco más este módulo con la funcionalidad de algunos de los botones por completo, sin embargo aun falta de agregar otras funciones que se desarrollaran en el siguiente sprint
PROG-05	90%	Se logró desarrollar y adecuar la funcionalidad de la base de datos para la funcionalidad del módulo pero aún quedan algunos puntos que se desarrollarán en el siguiente sprint.
PROG-06	90%	Se desarrolló un poco más el módulo, sin embargo aún faltan puntos que se desarrollaran en el siguiente sprint.
PROG-07	100%	Se desarrolló completamente y se conectó con la base de datos que muestra los distintos consejos.
PROG-08	80%	Se desarrolló la implementación backend del módulo, se seguirá el desarrollo en el siguiente sprint.
PROG-09	100%	Se completó la interfaz y se dio conexión a la base de datos mostrando las frases.

Artefacto	Aspecto a evaluar	Evaluación	Comentarios
PROG-04	90%	Programación	Se consiguió desarrollar un poco más este módulo, se seguirá el desarrollo en el siguiente sprint.
PROG-05	90%	Programación	Se desarrolló un poco más el módulo, sin embargo aún faltan puntos que se desarrollaran en el siguiente sprint.
PROG-06	90%	Programación	Se desarrolló un poco más el módulo, sin embargo aún faltan puntos que se desarrollaran en el siguiente sprint.
PROG-07	100%	Programación	Se desarrolló completamente y se conectó con la base de datos que muestra los distintos consejos.
PROG-08	80%	Programación	Se desarrolló el servicio para dar un seguimiento a la calidad del sueño con sensores, se seguirá el desarrollo en el siguiente sprint.
PROG-09	100%	Programación	Se completó la interfaz y se dio conexión a la base de datos mostrando las frases

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

MeditaciónActivity.java

```
package mx.GPS.healthec;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.content.res.Configuration;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;

public class MeditacionActivity extends AppCompatActivity {

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

        int orientation=getResources().getConfiguration().orientation;
        if(orientation== Configuration.ORIENTATION_PORTRAIT){
            getSupportActionBar().hide();
            getWindow().addFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN);
        }else
            getSupportActionBar().show();
    }

    public void btnSueño(View v){
        startActivity(new Intent(MeditacionActivity.this,ConciliacionSueno.class));
    }
}
```

Sonido.java

```
package mx.GPS.healthec;

public class Sonido {

    String nombre;
    String descripcion;
    String sonido;

    public Sonido(String nombre, String sonido, String descripcion) {
        this.nombre = nombre;
        this.descripcion=descripcion;
        this.sonido = sonido;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getSonido() {
        return sonido;
    }
}
```

```
public void setSonido(String sonido) {
    this.sonido = sonido;
}
}
```

ConciliacionSueno.java

```
package mx.GPS.healthec;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.res.Configuration;
import android.os.Bundle;
import android.view.WindowManager;

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

public class ConciliacionSueno extends AppCompatActivity {

    RecyclerView rvLista;
    List<Sonido> listaSonidos;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_conciliacion_sueno);

        int orientation=getResources().getConfiguration().orientation;
        if(orientation== Configuration.ORIENTATION_PORTRAIT){
            getSupportActionBar().hide();
            getWindow().addFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN);
        }else
            getSupportActionBar().show();

        rvLista=findViewById(R.id.rvLista);
        rvLista.setLayoutManager(new GridLayoutManager(this,1));

        listaSonidos=new ArrayList<>();
        listaSonidos.add(new Sonido("Sonidos de la naturaleza","sonidoUno","Los sonidos de la naturaleza, el canto de los pájaros, el sonido del mar y el viento en los árboles, pueden ayudar a reducir el estrés y promover la relajación."));
        listaSonidos.add(new Sonido("Musica suave","sonidoDos","La música suave y tranquila, como la música clásica, la música ambiental o la música de meditación, puede ayudarte a reducir la ansiedad y promover la relajación."));
        listaSonidos.add(new Sonido("ASMR","sonidoTres","Las sensaciones autónomas meridianas del cerebro (ASMR, por sus siglas en inglés) son un fenómeno que se produce cuando escuchamos sonidos suaves y repetitivos, como susurros, crujidos o golpeteos."));
        listaSonidos.add(new Sonido("Sonidos blandos","sonidoCuatro","Los sonidos blancos, como el sonido de un ventilador, de la lluvia o el sonido de la estática, pueden ayudar a bloquear otros sonidos y promover un ambiente tranquilo y relajante."));
        listaSonidos.add(new Sonido("Sonidos de instrumentos musicales","sonidoCinco","Los sonidos de instrumentos musicales como el piano, la flauta o el violín, pueden ser relajantes y promover la tranquilidad."));

        RecyclerView.Adapter adapter=new
        AdaptadorSonidos(ConciliacionSueno.this,listaSonidos);
        rvLista.setAdapter(adapter);
    }
}
```

AdaptadorSonidos.java

```
package mx.GPS.healthec;

import android.annotation.SuppressLint;
import android.content.Context;
import android.media.MediaPlayer;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
```



```
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.List;

public class AdaptadorSonidos extends
RecyclerView.Adapter<AdaptadorSonidos.SonidoViewHolder> {

    MediaPlayer mediaPlayer;
    Context context;
    List<Sonido> listaSonidos;

    public AdaptadorSonidos(Context context, List<Sonido> listaSonidos) {
        this.context = context;
        this.listaSonidos = listaSonidos;
    }

    @NonNull
    @Override
    public AdaptadorSonidos.SonidoViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View v=
LayoutInflater.from(parent.getContext()).inflate(R.layout.list_element,parent,false);
        return new SonidoViewHolder(v);
    }

    @Override
    public void onBindViewHolder(final AdaptadorSonidos.SonidoViewHolder holder,
@SuppressLint("RecyclerView") final int position) {
        holder.tvNombre.setText(listaSonidos.get(position).getNombre());
        holder.tvDesc.setText(listaSonidos.get(position).getDescripcion());
        holder.btnReproducir.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (mediaPlayer!=null){
                    mediaPlayer.stop();
                }
                if (listaSonidos.get(position).getSonido().equals("sonidoUno")){
                    mediaPlayer=MediaPlayer.create(context,R.raw.naturaleza);
                    mediaPlayer.start();
                }else if(listaSonidos.get(position).getSonido().equals("sonidoDos")) {
                    mediaPlayer = MediaPlayer.create(context, R.raw.meditacion);
                    mediaPlayer.start();
                }
                else if(listaSonidos.get(position).getSonido().equals("sonidoTres")) {
                    mediaPlayer = MediaPlayer.create(context, R.raw.asmr);
                    mediaPlayer.start();
                }
                else if(listaSonidos.get(position).getSonido().equals("sonidoCuatro")) {
                    mediaPlayer = MediaPlayer.create(context, R.raw.lluvia);
                    mediaPlayer.start();
                }
                else if(listaSonidos.get(position).getSonido().equals("sonidoCinco")) {
                    mediaPlayer = MediaPlayer.create(context, R.raw.piano);
                    mediaPlayer.start();
                }
            }
        });
        holder.btnDetener.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mediaPlayer.stop();
            }
        });
    }

    @Override
    public int getItemCount() {
        return listaSonidos.size();
    }

    public class SonidoViewHolder extends RecyclerView.ViewHolder {

        ImageView icon;
```

```

        TextView tvNombre, tvDesc;
        Button btnReproducir, btnDetener;

        public SonidoViewHolder(View itemView) {
            super(itemView);

            icon=itemView.findViewById(R.id.icon);
            tvNombre=itemView.findViewById(R.id.tvNombre);
            tvDesc=itemView.findViewById(R.id.descripcion);
            btnDetener=itemView.findViewById(R.id.btnDetener);
            btnReproducir=itemView.findViewById(R.id.btnReproducir);
        }
    }
}

```

activity_meditacion.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="@drawable/fondomed"
    android:backgroundTint="#000000"
    android:orientation="vertical"
    tools:context=".MeditacionActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/fondomed"
        android:orientation="vertical">

        <Space
            android:layout_width="match_parent"
            android:layout_height="284dp" />

        <Button
            android:id="@+id/btnSueno"
            android:layout_width="340dp"
            android:layout_height="50dp"
            android:layout_marginLeft="35dp"
            android:backgroundTint="#F7E8D5"
            android:backgroundTintMode="add"
            android:foregroundTint="#FFF7F7"
            android:onClick="btnSueño"
            android:text="Conciliacion del sueño"
            android:textColor="#908883"
            android:textIsSelectable="false"
            android:textSize="16sp"
            android:textStyle="bold"
            app:rippleColor="#FEF5E7"
            tools:ignore="MissingConstraints" />

        <Space
            android:layout_width="match_parent"
            android:layout_height="25dp" />

        <Button
            android:id="@+id/button7"
            android:layout_width="345dp"
            android:layout_height="50dp"
            android:layout_marginLeft="35dp"
            android:backgroundTint="#F7E8D5"
            android:backgroundTintMode="add"
            android:foregroundTint="#FFF7F7"
            android:text="ALIVIO DE ANSIEDAD"
            android:textColor="#908883"
            android:textSize="16sp"
            android:textStyle="bold"
            app:rippleColor="#FEF5E7"
            tools:ignore="MissingConstraints" />
    </LinearLayout>
</LinearLayout>

```

```
<Space
    android:layout_width="match_parent"
    android:layout_height="30dp" />

<Button
    android:id="@+id/button8"
    android:layout_width="345dp"
    android:layout_height="50dp"
    android:layout_marginLeft="35dp"
    android:backgroundTint="#F7E8D5"
    android:backgroundTintMode="add"
    android:text="RELAJACION VESPERTINA"
    android:textColor="#908883"
    android:textSize="16sp"
    android:textStyle="bold"
    app:rippleColor="#FEF5E7"
    tools:ignore="MissingConstraints" />

<Space
    android:layout_width="match_parent"
    android:layout_height="25dp" />

<Button
    android:id="@+id/button9"
    android:layout_width="345dp"
    android:layout_height="50dp"
    android:layout_marginLeft="35dp"
    android:backgroundTint="#F7E8D5"
    android:backgroundTintMode="add"
    android:text="ALIVIO DE SOBREPENSAMIENTO"
    android:textColor="#908883"
    android:textSize="16sp"
    android:textStyle="bold"
    app:rippleColor="#FEF5E7"
    tools:ignore="MissingConstraints" />
</LinearLayout>

</LinearLayout>
```

activity_conciliacion_sueño.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="@drawable/sueno"
    tools:context=".ConciliacionSueno">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvLista"
        android:layout_width="match_parent"
        android:layout_height="647dp"
        android:layout_marginTop="90dp" />
</LinearLayout>
```

list_element.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    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/cv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:animateLayoutChanges="true"
    app:cardBackgroundColor="#65645251"
    app:cardCornerRadius="20dp"
    app:cardElevation="4dp"
    app:cardUseCompatPadding="true">

    <LinearLayout
        android:layout_width="match_parent"
```

```

android:layout_height="wrap_content"
android:animateLayoutChanges="true"
android:orientation="vertical">

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#65645251"
    android:padding="6dp">

    <ImageView
        android:id="@+id/icon"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_centerVertical="true"
        android:layout_gravity="center"
        android:paddingLeft="14dp"
        android:paddingRight="14dp"
        app:srcCompat="@drawable/music"
        app:tint="#FFFFFF"></ImageView>

    <LinearLayout
        android:id="@+id/linearlayout"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_marginStart="4dp"
        android:layout_marginLeft="4dp"
        android:layout_marginTop="2dp"
        android:layout_toEndOf="@+id/icon"
        android:layout_toRightOf="@+id/icon"
        android:orientation="vertical">

        <TextView
            android:id="@+id/tvNombre"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="2dp"
            android:text="Sonido"
            android:textColor="@color/white"
            android:textSize="24sp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/descripcion"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="descripcion"
            android:textColor="#FFFFFF"
            android:textSize="18sp" />

        <Space
            android:layout_width="match_parent"
            android:layout_height="10dp" />

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

            <Button
                android:id="@+id/btnReproducir"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:backgroundTint="#303034"
                android:text="Reproducir"
                android:textColor="#FFFFFF"
                tools:ignore="TouchTargetSizeCheck" />

            <Space
                android:layout_width="58dp"
                android:layout_height="wrap_content" />

            <Button

```

```

        android:id="@+id/btnDetener"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:backgroundTint="#303034"
        android:text="Detener"
        android:textColor="#FAFCFF" />

    </TableRow>

</LinearLayout>

</RelativeLayout>

</LinearLayout>

</androidx.cardview.widget.CardView>

```

Anexo B.

PROG-05 Programar activity de Recetas

ActivityRecetas.java

```

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;

import org.checkerframework.checker.units.qual.A;

import java.util.List;

public class RecetasActivity extends AppCompatActivity {
    //Constantes
    ListView lv_recetasList;
    ArrayAdapter recetasArrayAdapter;
    private int [] idRecetas;
    private String [] recetasTitulo;
    private String [] recetasPasos;
    private String [] recetasIngredientes;
    private int [] recetasImagen;

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

        //ListView
        lv_recetasList = findViewById(R.id.lv_recetas);

        //Llenado del ViewList
        db = new DataBaseHealthec(RecetasActivity.this);
        actualizarList();
        lv_recetasList = findViewById(R.id.lv_recetas);
        MiAdaptador adaptador = new MiAdaptador( this, recetasTitulo, recetasImagen );
        lv_recetasList.setAdapter(adaptador);
    }

    public void actualizarList (){
        List<RecetasModel> modeloRecetasList = db.getRecetas();
        idRecetas = new int [modeloRecetasList.size()];
        recetasTitulo = new String [modeloRecetasList.size()];
        recetasPasos = new String [modeloRecetasList.size()];
    }
}

```

```

recetasIngredientes = new String [modeloRecetasList.size()];
recetasImagen = new int[modeloRecetasList.size()];
for (int i = 0; i < modeloRecetasList.size(); i++){
    RecetasModel modeloRecetas = modeloRecetasList.get(i);
    idRecetas[i] = modeloRecetas.getIdReceta();
    recetasTitulo[i] = modeloRecetas.getNombreReceta();
    recetasPasos[i] = modeloRecetas.getPasosReceta();
    recetasIngredientes[i] = modeloRecetas.getIngredientesReceta();
    recetasImagen[i] = modeloRecetas.getImagenReceta();
}
}

class MiAdaptador extends ArrayAdapter{
    private Context context;
    private String [] recetaTitulo;
    private String [] recetaPasos;
    private int [] recetasImagen;
    public MiAdaptador(Context c, String [] recetaTitulo, int [] recetasImagen ) {

        super(c, R.layout.list_recetas, R.id.txtvTituloReceta, recetaTitulo );
        context = c;
        this.recetaTitulo = recetaTitulo;
        this.recetasImagen = recetasImagen;
    }

    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup
parent) {
        if(convertView == null){
            LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            //LayoutInflater inflater = LayoutInflater (context)
            convertView = inflater.inflate(R.layout.list_recetas, parent, false);
        }
        ImageView imagen = convertView.findViewById(R.id.imgvRecetas);
        TextView titulo = convertView.findViewById(R.id.txtvTituloReceta);

        imagen.setImageResource( recetasImagen[position] );
        titulo.setText( recetasTitulo[position] );
        return super.getView(position, convertView, parent);
    }
}
}

```

RecetasModel.java

```

package mx.GPS.healthec;

public class RecetasModel {
    private int idReceta;

    private String nombreReceta;

    private String pasosReceta;

    private String ingredientesReceta;
    private int imagenReceta;

    public RecetasModel(int idReceta, String nombreReceta, String pasosReceta, String
ingredientesReceta, int imagenReceta) {
        this.idReceta = idReceta;
        this.nombreReceta = nombreReceta;
        this.pasosReceta = pasosReceta;
        this.ingredientesReceta = ingredientesReceta;
        this.imagenReceta = imagenReceta;
    }

    public RecetasModel() {
    }

    @Override
    public String toString() {

```

```
        return "MenuModel{" +
            "idReceta=" + idReceta +
            ", nombreReceta='" + nombreReceta + '\'' +
            ", pasosReceta='" + pasosReceta + '\'' +
            ", ingredientesReceta='" + ingredientesReceta + '\'' +
            '}' +
        }

    public int getIdReceta() {
        return idReceta;
    }

    public void setIdReceta(int idReceta) {
        this.idReceta = idReceta;
    }

    public String getNombreReceta() {
        return nombreReceta;
    }

    public void setNombreReceta(String nombreReceta) {
        this.nombreReceta = nombreReceta;
    }

    public String getPasosReceta() {
        return pasosReceta;
    }

    public void setPasosReceta(String pasosReceta) {
        this.pasosReceta = pasosReceta;
    }

    public String getIngredientesReceta() {
        return ingredientesReceta;
    }

    public void setIngredientesReceta(String ingredientesReceta) {
        this.ingredientesReceta = ingredientesReceta;
    }

    public int getImagenReceta() {
        return imagenReceta;
    }

    public void setImagenReceta(int imagenReceta) {
        this.imagenReceta = imagenReceta;
    }
}
```

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/btnInformacion"
        android:layout_width="160dp"
        android:layout_height="80dp"
        android:layout_marginTop="48dp"
        android:background="#F6DDB9"
        android:text="@string/mi_informacion"
        android:textColor="#616161"
        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="@string/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/ibtnAtras"
        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"
        tools:ignore="SpeakableTextPresentCheck, ImageContrastCheck"
        android:contentDescription="@string/back" />

    <Button
        android:id="@+id/btnCitas"
        android:layout_width="160dp"
        android:layout_height="80dp"
```



```

android:background="#F6DDB9"
android:text="@string/citas"
android:textColor="#616161"
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"
tools:ignore="VisualLintOverlap" />

```

```

<Button
    android:id="@+id/btnMedicamentos"
    android:layout_width="160dp"
    android:layout_height="80dp"
    android:background="#F6DDB9"
    android:text="@string/medicamentos"
    android:textColor="#616161"
    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"
    tools:ignore="TextSizeCheck,VisualLintOverlap" />

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="284dp"
    android:orientation="vertical"
    app:layout_constraintTop_toTopOf="parent"
    tools:layout_editor_absoluteX="-27dp">

```

```

<TextView
    android:id="@+id/textView10"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/recordatorios2"
    android:textSize="20sp"
    tools:ignore="TextSizeCheck" />

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

```

```

<EditText
    android:id="@+id/txtRecordatorio"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:ems="10"
    android:hint="@string/nuevo_recordatorio"
    android:inputType="textPersonName"
    tools:ignore="TouchTargetSizeCheck,VisualLintTextFieldSize,VisualLintBounds"
    android:autofillHints="" />

```

```

<TextView
    android:id="@+id/btnAgregar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="center"

```

```

        android:text="@string/mas"
        android:textSize="30sp" />

    </LinearLayout>

    <ListView
        android:id="@+id/lista"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1" />
    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

RecordatoriosActivity.java

```

package mx.GPS.healthec;

import androidx.activity.result.contract.ActivityResultContracts;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ListView;
import android.widget.TableLayout;
import android.widget.TextView;
import android.widget.Toast;

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

public class RecordatoriosActivity extends AppCompatActivity implements
AdapterView.OnItemClickListener {
    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", "Clonasepam",
"Ketorolako"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_recordatorios);
        agregaractividad=findViewById(R.id.btnAgregar);
        actividad=findViewById(R.id.txtRecordatorio);
        list = findViewById(R.id.lista);
        list.setOnItemClickListener(this);
    }

```

```

mAdapter = new ListAdapter(RecordatoriosActivity.this, R.layout.item_row,mLista);
list.setAdapter(mAdapter);
agregaractividad.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        mLista.add(new Modelo(list.getCount()+1+"",actividad.getText().toString()));
        mAdapter= new ListAdapter(RecordatoriosActivity.this,
R.layout.item_row,mLista);
        list.setAdapter(mAdapter);
        actividad.setText("");
    }
});

// Enlazar vistas con variables
miInfoBtn = findViewById(R.id.btnInformacion);
citasBtn = findViewById(R.id.btnCitas);
medicamentosBtn = findViewById(R.id.btnMedicamentos);
menuBtn = findViewById(R.id.ibtnAtras);
titleLabel = findViewById(R.id.textView2);

// 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();
        }
    });
}

private ListView list;
private TextView agregaractividad;
private EditText actividad;
private List<Modelo>mLista= new ArrayList<>();
Modelo modelo;
ListAdapter mAdapter;

@Override
public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
    Toast.makeText(this, "Actividad seleccionada: "+i,Toast.LENGTH_SHORT).show();
    Intent intent = new Intent(this, SegundaActivity.class);
    intent.putExtra("nombre", mAdapter.getItem(i).getNombreactividad());
    startActivityForResult(intent,l);
}

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data)
{
    super.onActivityResult(requestCode,resultCode,data);
    if(requestCode==1 && resultCode==RESULT_OK)
    {
    }
}
}

class Modelo {

    private String prioridadactividad;
    private String nombreactividad;

    public Modelo() {
    }

    public Modelo(String prioridadactividad, String nombreactividad) {
        this.prioridadactividad = prioridadactividad;
        this.nombreactividad = nombreactividad;
    }

    public String getPrioridadactividad() {
        return prioridadactividad;
    }

    public void setPrioridadactividad(String prioridadactividad) {
        this.prioridadactividad = prioridadactividad;
    }

    public String getNombreactividad() {
        return nombreactividad;
    }

    public void setNombreactividad(String nombreactividad) {
        this.nombreactividad = nombreactividad;
    }
}

```

```

    }
}

class ListAdapter extends ArrayAdapter <Modelo> {
    private List<Modelo> miLista;
    private Context mContext;
    private int resourceLayout;
    public ListAdapter(@NonNull Context context, int resource, List<Modelo> objects) {
        super(context, resource);
        this.miLista=objects;
        this.mContext=context;
        this.resourceLayout=resource;
    }

    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent)
    {
        View view = convertView;
        if(view==null)
            view= LayoutInflater.from(mContext).inflate(R.layout.item_row,null);

        Modelo modelo=miLista.get(position);
        TextView texttoact= view.findViewById(R.id.txtactividad);
        texttoact.setText(modelo.getNombreactividad());
        TextView textoprioridad=view.findViewById(R.id.txtprioridad);
        textoprioridad.setText(modelo.getPrioridadactividad());
        return view;
    }
}

```

activity_segunda.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:orientation="vertical"
    tools:context=".SegundaActivity">

    <TextView
        android:id="@+id/textViewNombre"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/textview" />
</LinearLayout>

```

SegundaActivity.java

```

package mx.GPS.healthec;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;

public class SegundaActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_segunda);
        TextView textView=findViewById(R.id.textViewNombre);
    }
}

```

```
        textView.setText(getIntent().getStringExtra("nombre"));
    }
}
```

item_row.java

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    tools:context=".item_row">

    <TextView
        android:id="@+id/txtprioridad"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/textview"
        tools:ignore="VisualLintOverlap" />

    <TextView
        android:id="@+id/txtactividad"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/textview"
        android:layout_marginStart="17dp"
        android:layout_toEndOf="@id/txtprioridad"
        tools:ignore="UnknowId"/>
</RelativeLayout>
```

item_row.xml

```
package mx.GPS.healthec;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class item_row extends AppCompatActivity {

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



Anexo D.

PROG-07 Programar activity de Consejos saludables

ConsejosSaludablesActivity.java

```
package mx.GPS.healthec;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
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;

import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class ConsejosSaludables extends AppCompatActivity {

    TextView txtvConsejo;

    ImageView imgvConsejo;

    ProgressBar prgsbImagenes;

    private DatabaseReference mDataBase;

    static int idSwitch=1;
    static int progress = 0;

    static int cont=1;
    @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);

        mDataBase = FirebaseDatabase.getInstance().getReference();

        mDataBase.child("consejos").addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {

                if(snapshot.exists()){
                    // String consejo = snapshot.getValue().toString();
                    String consejo = snapshot.child(cont+"").getValue().toString();
                    txtvConsejo.setText("Consejo :  "+ consejo);
                }

            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }

        });

    }
}
```

```
public void btnSiguienteClick (View v){

    cont+=1;

    if(cont >= 12)
        cont = 1;

    mDataBase.child("consejos").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {

            if(snapshot.exists()){
                // String consejo = snapshot.getValue().toString();
                String consejo = snapshot.child(cont+"").getValue().toString();
                txtvConsejo.setText("Consejo : "+ consejo);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }

        });

    progress+=10;
    idSwitch++;

    prgsbImagenes.setProgress(progress,true);

    if(progress==110){
        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;

    }

}

}
```


Healthec



Consejo : Las dietas muy bajas en calorías,
empeoran el rendimiento y pueden ser
perjudiciales para la salud.

SIGUIENTE

Anexo E.

PROG-08 Programar activity de Información de sueño diario

AcelerometerService.java

```
package mx.GPS.healthec;

import android.app.AlarmManager;
import android.app.PendingIntent;
import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.IBinder;
import android.os.Handler;

import androidx.annotation.Nullable;

import java.util.Calendar;

public class AcelerometerService extends Service implements SensorEventListener {
    private SensorManager sensorManager;
    private Sensor accelerometer;
    private boolean isServiceRunning = false;
    private Handler handler;
    private int serviceDurationSeconds;
    private int startHour;

    private Runnable stopServiceRunnable = new Runnable() {
        @Override
        public void run() {
            stopSelf();
        }
    };

    @Override
    public void onCreate() {
        super.onCreate();
        sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
        accelerometer = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        handler = new Handler();
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        if (!isServiceRunning) {
            serviceDurationSeconds = intent.getIntExtra("duration_seconds", 0);
            startHour = intent.getIntExtra("start_hour", 0);
            startAccelerometer();
            handler.postDelayed(stopServiceRunnable, serviceDurationSeconds * 1000L);
            isServiceRunning = true;
        }
        return START_STICKY;
    }

    private void startAccelerometer() {
        sensorManager.registerListener(this, accelerometer,
        SensorManager.SENSOR_DELAY_NORMAL);
    }

    private void stopAccelerometer() {
        sensorManager.unregisterListener(this);
    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        // Lógica para procesar los datos del acelerómetro
        if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
            float x = event.values[0];
            float y = event.values[1];
        }
    }
}
```

```

        float z = event.values[2];

        //Aquí puedes hacer algo con los valores del acelerómetro, como mostrarlos en
        una vista TextView
        TextView textView = findViewById(R.id.textView);
        textView.setText("X: " + x + "\nY: " + y + "\nZ: " + z);
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {
        // No es necesario implementar este método en este ejemplo
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        stopAccelerometer();
        handler.removeCallbacks(stopServiceRunnable);
    }

    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }

    public static void scheduleService(Context context) {
        Intent intent = new Intent(context, AcelerometerService.class);

        PendingIntent startServicePendingIntent = PendingIntent.getService(context, 0,
        intent, PendingIntent.FLAG_UPDATE_CURRENT);

        PendingIntent stopServicePendingIntent = PendingIntent.getService(context, 1,
        intent, PendingIntent.FLAG_UPDATE_CURRENT);

        Calendar calendar = Calendar.getInstance();
        calendar.setTimeInMillis(System.currentTimeMillis());
        calendar.set(Calendar.HOUR_OF_DAY, startHour);
        calendar.set(Calendar.MINUTE, 0);
        calendar.set(Calendar.SECOND, 0);

        AlarmManager alarmManager = (AlarmManager)
        context.getSystemService(Context.ALARM_SERVICE);

        alarmManager.setInexactRepeating(AlarmManager.RTC_WAKEUP,
        calendar.getTimeInMillis(),
        AlarmManager.INTERVAL_DAY, startServicePendingIntent);

        calendar.add(Calendar.SECOND, durationSeconds);
        alarmManager.set(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(),
        stopServicePendingIntent);
    }
}

```

Anexo F.

PROG-09 Programar menu activity

menu_activity.java

```
android:layout_marginStart="130dp"
package mx.GPS.healthec;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.Random;

public class MenuActivity extends AppCompatActivity {

    private ImageView perfil;

    private TextView txtvFrase;

    private DatabaseReference mDataBase;

    private Random azar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menu);
        perfil = findViewById(R.id.imgvPerfil);
        perfil.setImageResource(R.drawable.saludconsejo);
        txtvFrase = findViewById(R.id.txtvFrase);

        mDataBase = FirebaseDatabase.getInstance().getReference();

        azar = new Random();

        int fraseAzar = azar.nextInt(10) + 1;

        mDataBase.child("frases").addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {

                if(snapshot.exists()){
                    // String consejo = snapshot.getValue().toString();
                    String frase = snapshot.child(fraseAzar+"").getValue().toString();
                    txtvFrase.setText(frase);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
    }
}
```

```
public void btnModuloSueñoClick (View v){
    Intent intent = new Intent(MenuActivity.this, SleepActivity.class);
    startActivity(intent);
}

public void btnModuloRecetasClick (View v){
    startActivity(new Intent( MenuActivity.this, RecetasActivity.class));
}

public void btnModuloRecordatorioClick(View v){
    startActivity(new Intent( MenuActivity.this, RecordatoriosActivity.class));
}

public void btnModuloMeditacionClick(View v){
    startActivity(new Intent( MenuActivity.this, MeditacionActivity.class));
}
}
```



Anexo G.

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	4
		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 realizar una función.	Concisión: Nivel de optimización de código en las funciones de la app.	4
		Eficiencia de ejecución: Nivel de eficiencia de ejecución en las funciones de la app.	4
		Operatividad: Nivel de facilidad de operación en las	3

		funciones de la app.	
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.	3
		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 incluso ajeno al proyecto.	4

	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.

Menú: Los menús son un componente común de la interfaz de usuario en muchos tipos de aplicaciones.