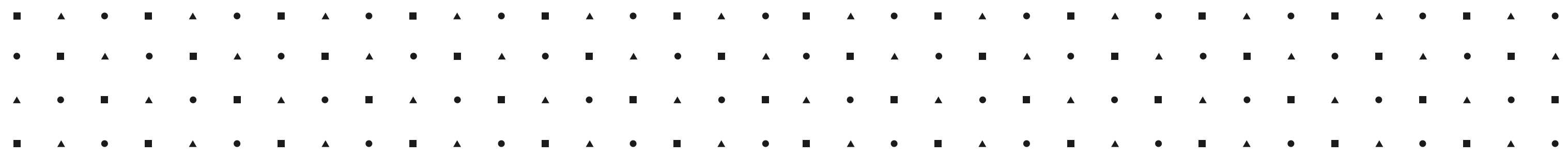
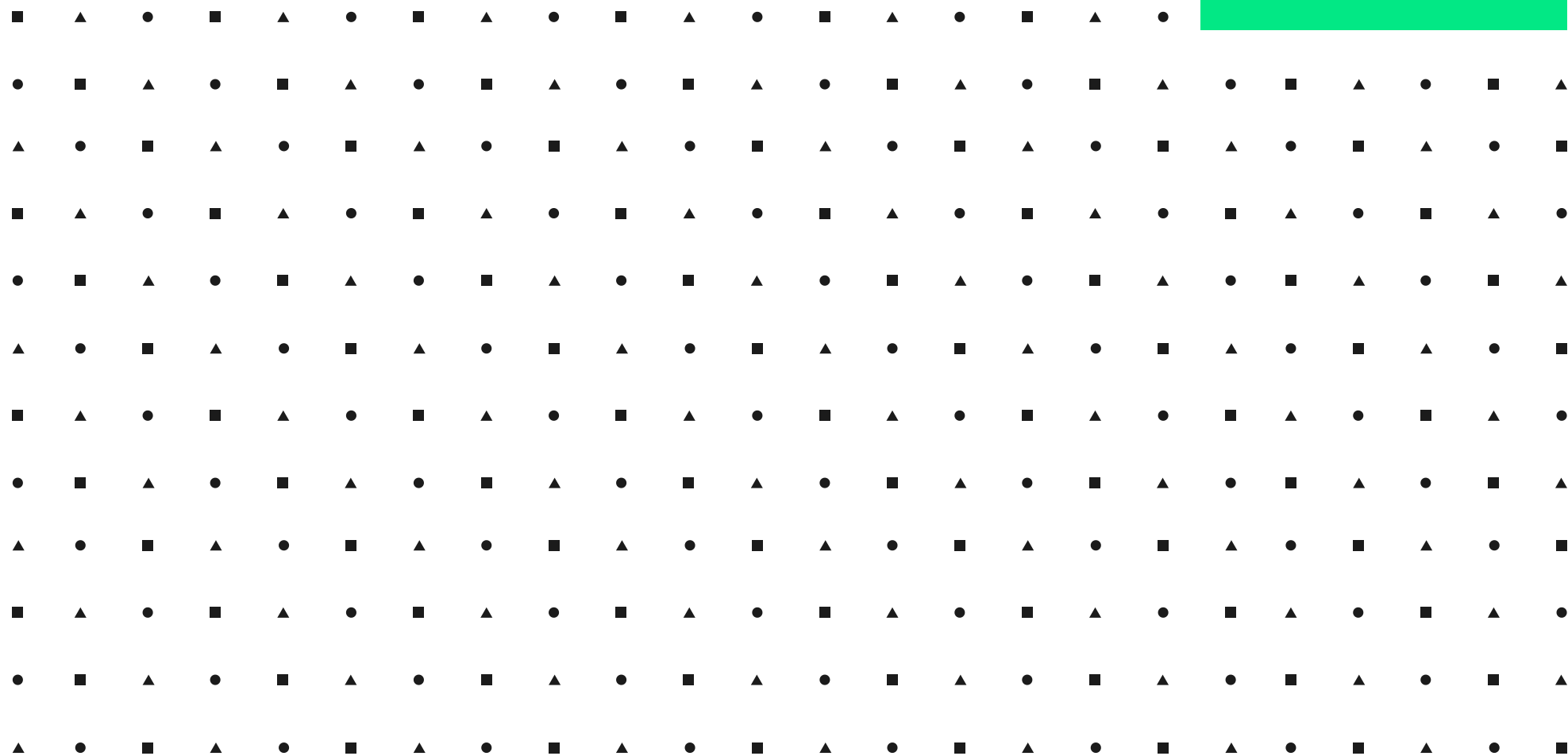


Manual Dental Clinic



01



¡Te damos la bienvenida!

Estamos encantados de tenerte con nosotros.

En esta ocasión te mostraremos el manual tecnico de nuestra app móvil "Dental Clinic"

Intergrantes:

Christian Alberto Batres Alemán

Sergio Andres Argueta Motto

Dairo Jonathan Fuentes Galdamez

Manuel Oswaldo Hernandez Lopez

Rene Alexander Carrillo Garcia

03

Introduccion

La implementación de tecnologías en el ámbito laboral muestra una tendencia alta, ya que facilita realizar diversas actividades con una acción que requiere muy poco esfuerzo, pero realiza un gran trabajo, si retomamos la forma habitual de hacer dicha tarea observaremos la eficiencia y efectividad que nos proporciona hacer uso de tecnologías. El desarrollo e implementación de software en plataformas móviles se ha convertido en una alternativa viable para realizar tareas en el ámbito laboral, ya que es un medio versátil, fácil uso y atractivo para el usuario. El proyecto que se generará para la materia DSM, será una aplicación que contara con dos partes un frontend que será realizada con el IDE Android Studio desarrollada en el lenguaje Java y utilizando distintas librerías que facilitaran dicho desarrollo y un backend en Firebase BAAS (Backend as a Service), creada con Google, esto nos permite que Google se encargue de crear y administras las funciones, base de datos e infraestructura de nuestra aplicación, del lado del desarrollador solo hace falta realizar distintas configuraciones en su plataforma firebase.google.com.

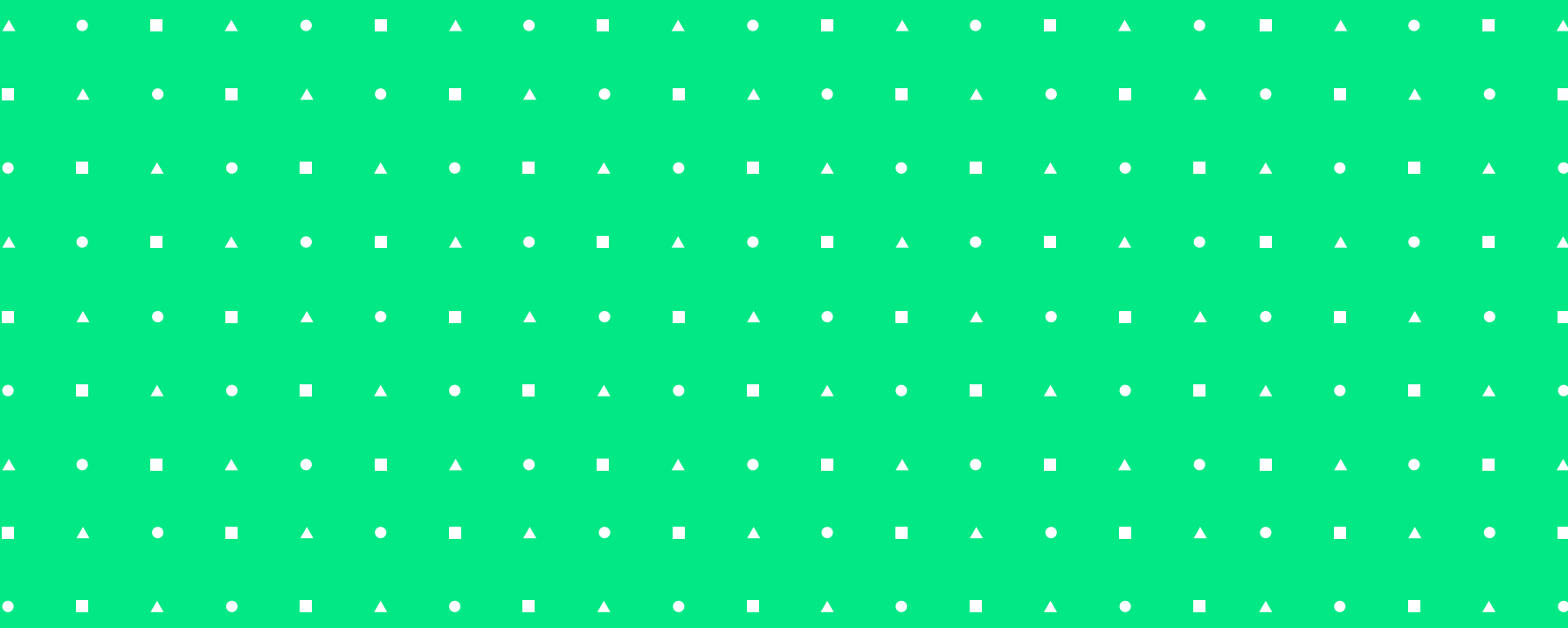


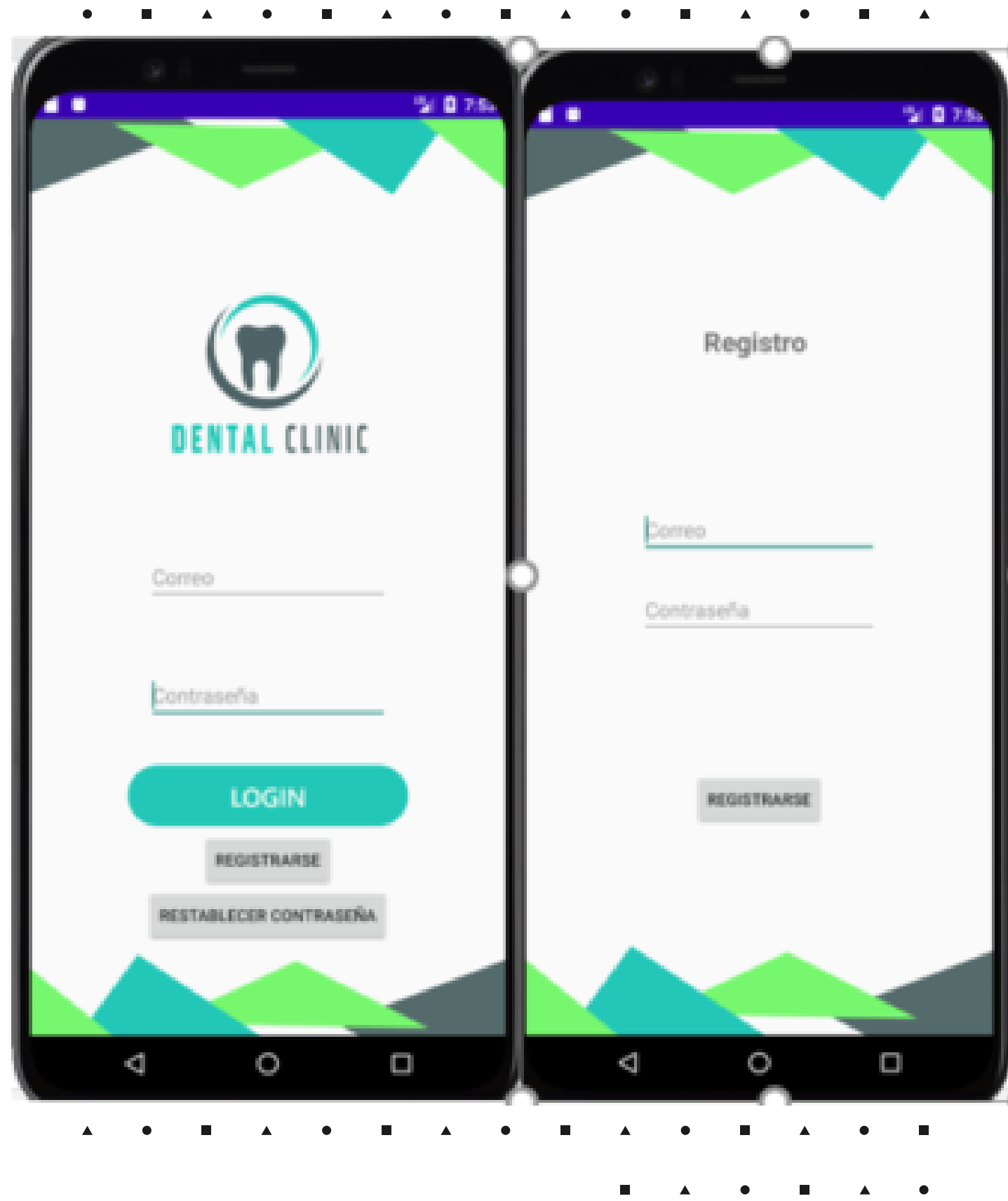
Objetivos

Desarrollar una aplicación móvil para una clínica dental a la cual se le facilite la gestión de clientes, médicos, calendario, suministros, etc. la cual será utilizada por el personal de dicho lugar para tener un mejor control en una plataforma desarrollada para dispositivos Android con un entorno amigable con el usuario y de fácil uso.

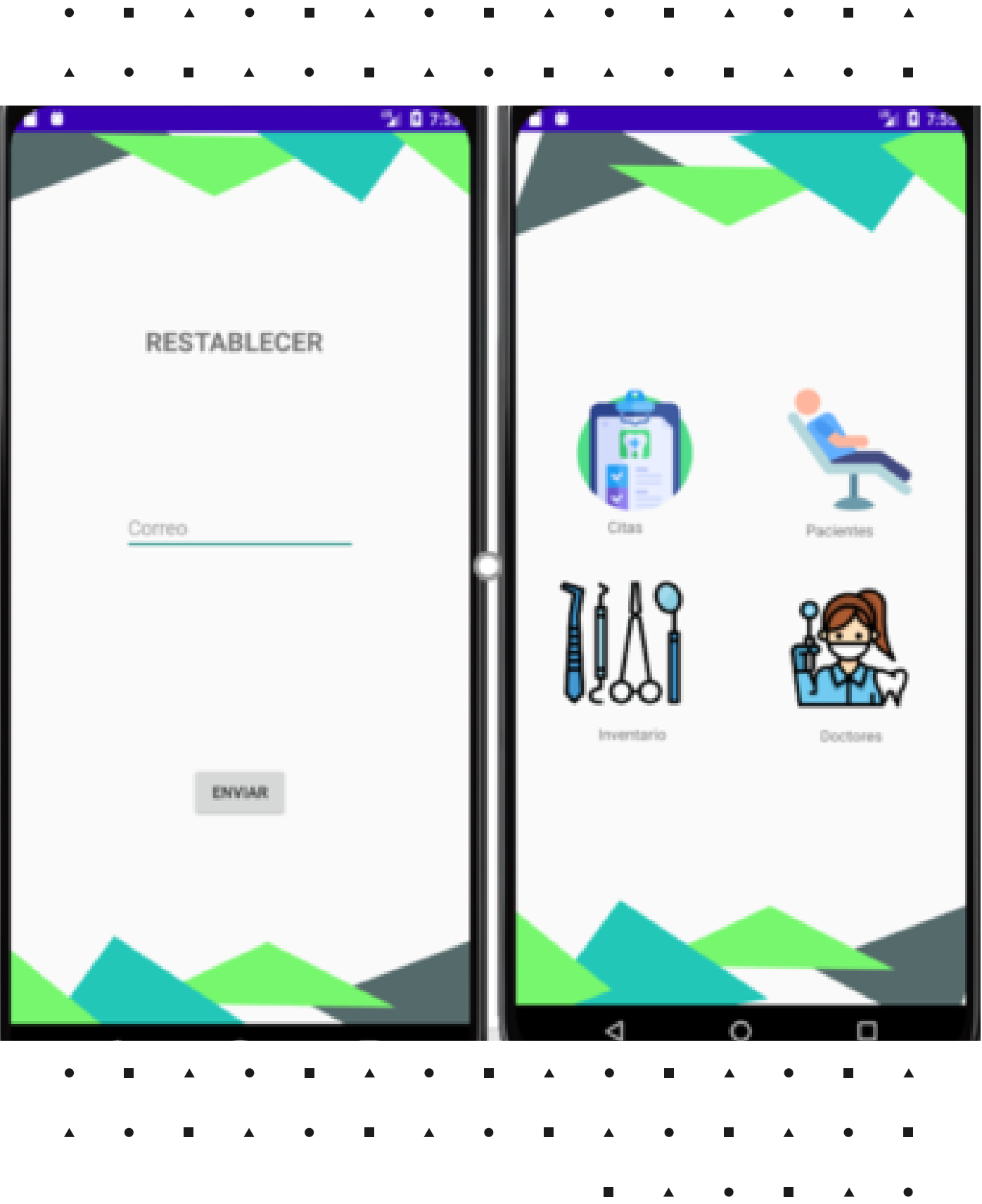
La aplicación deberá contar con las diferentes características:

- Implementación de un login para los diferentes tipos de usuarios que puede tener la aplicación.
- Realización y gestión desde el apartado de citas.
- Expediente clínico para cada cliente que podrá ser visto por los doctores.
- Inventario de los distintos insumos con los que cuenta la clínica.
- Interfaz Intuitiva (UX, UI)

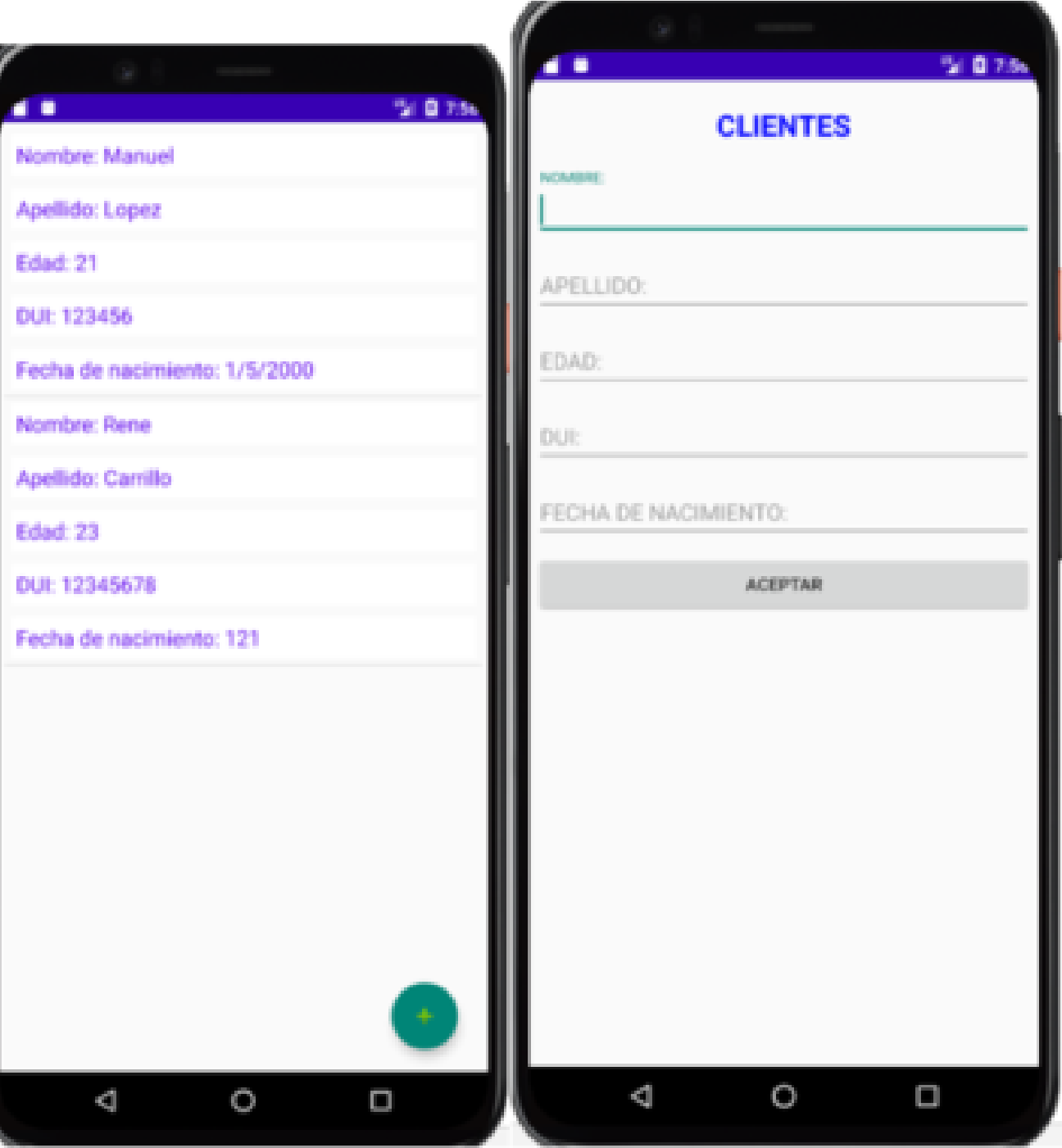
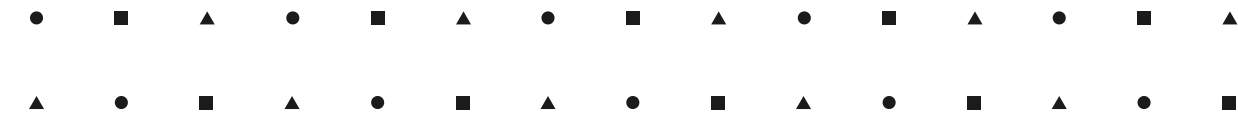




Login de Usuario

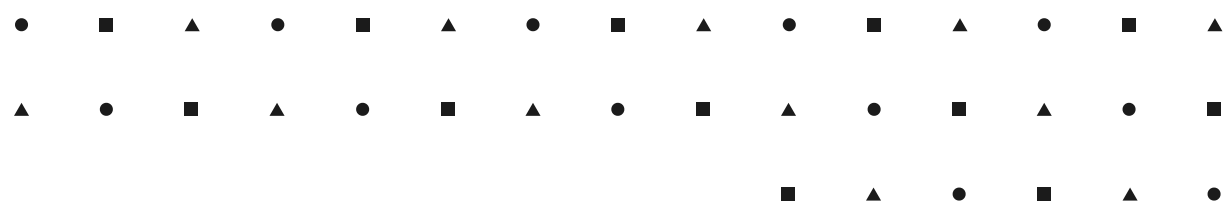
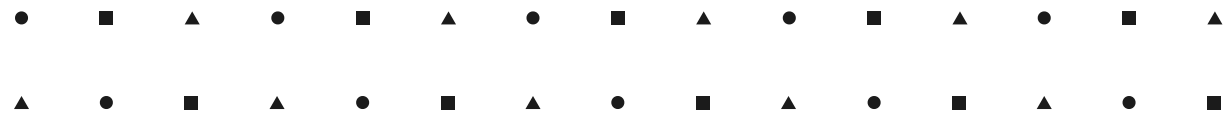


Restablecer Contraseña

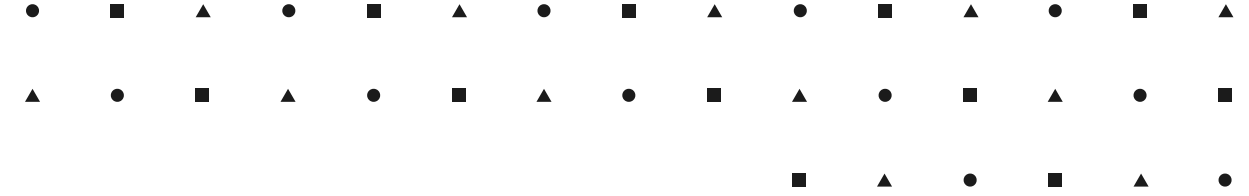
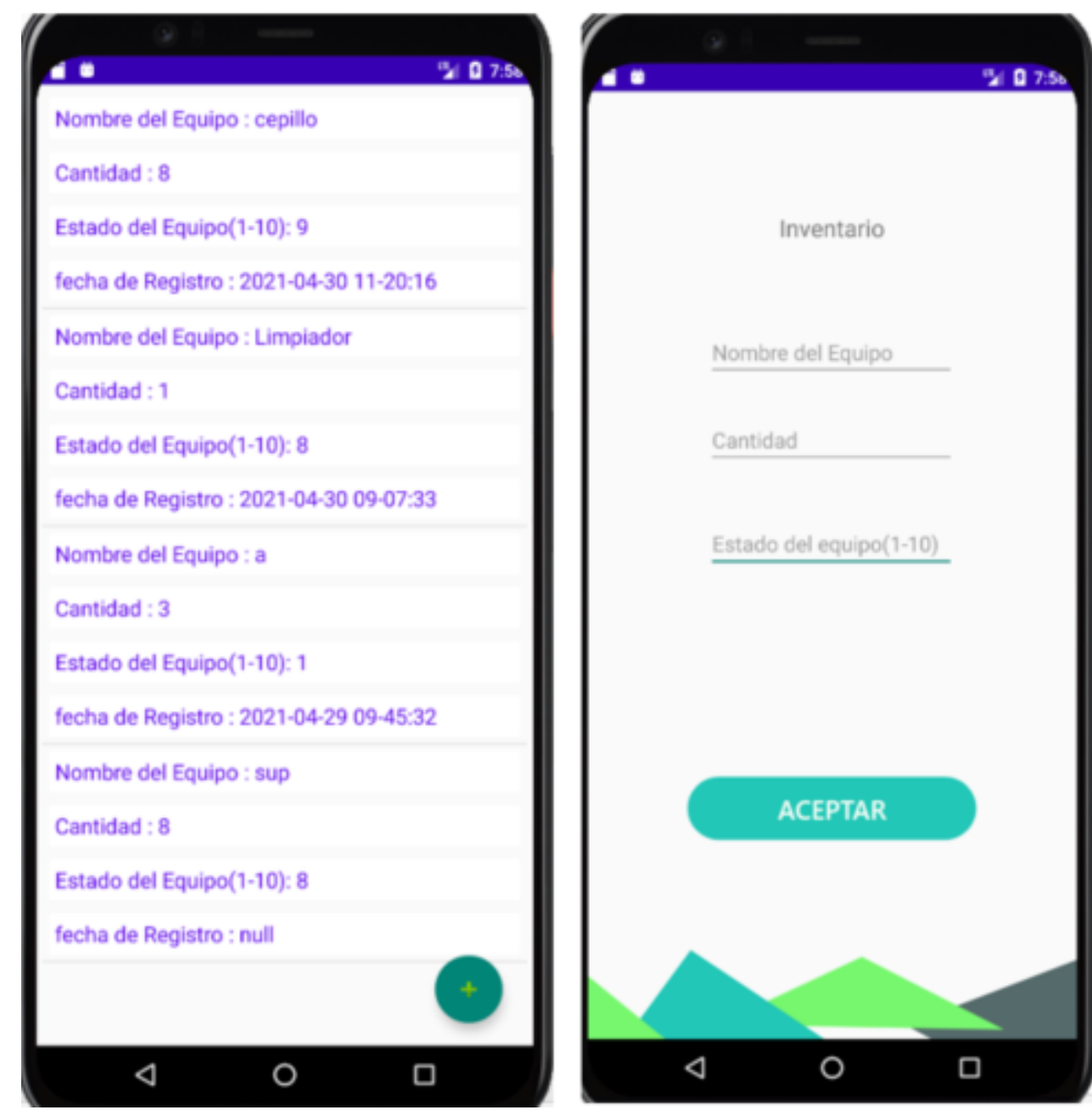
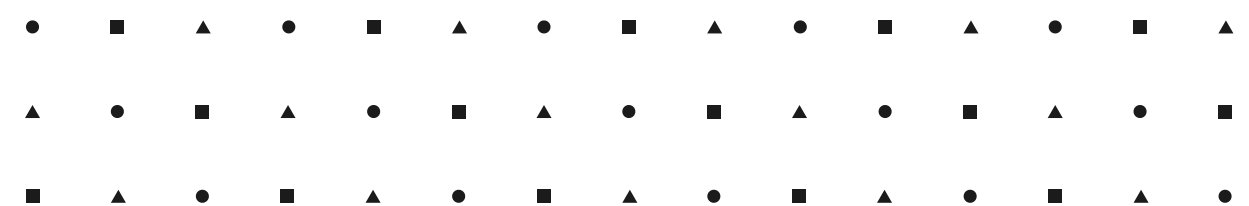


Cientes

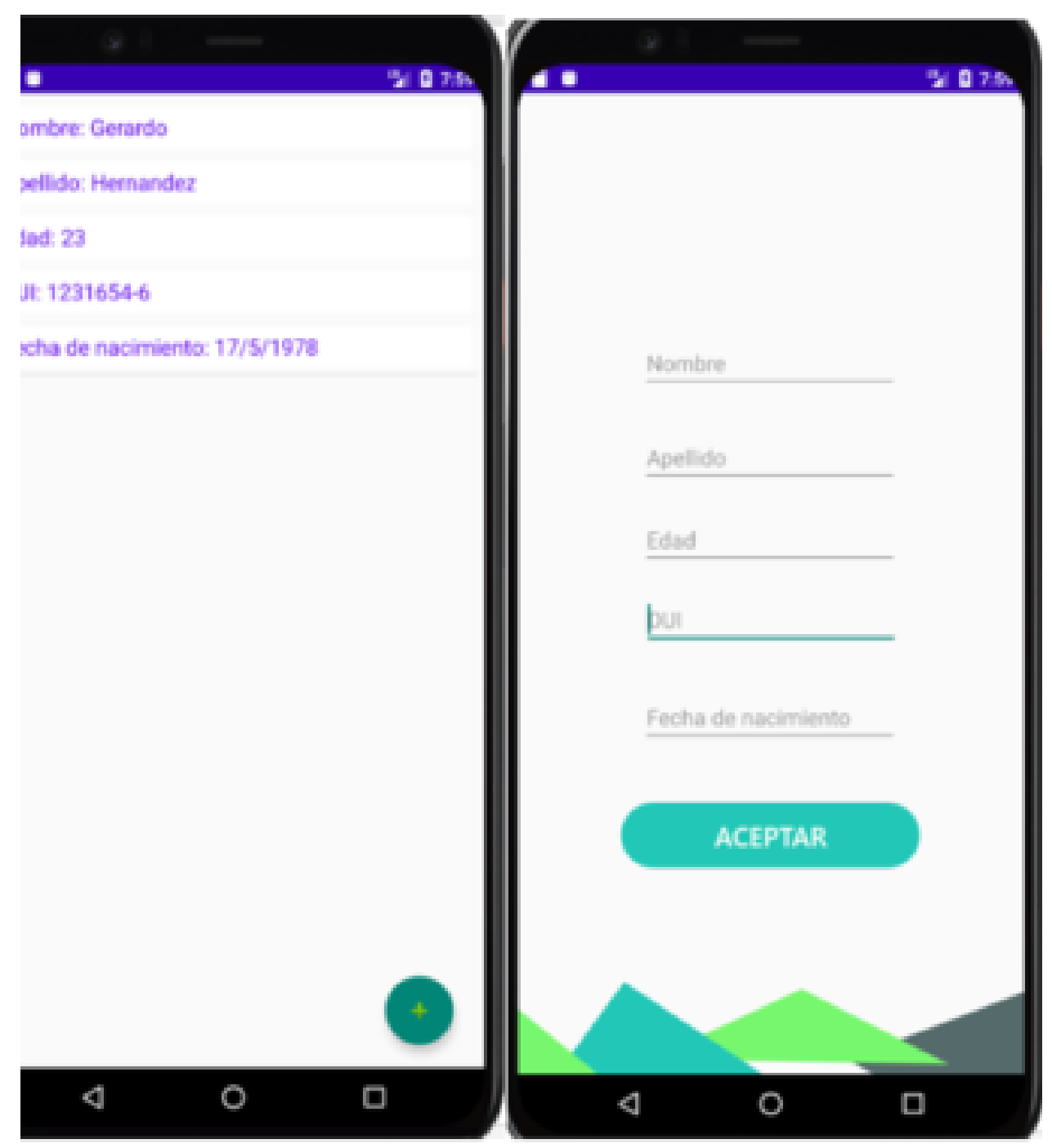
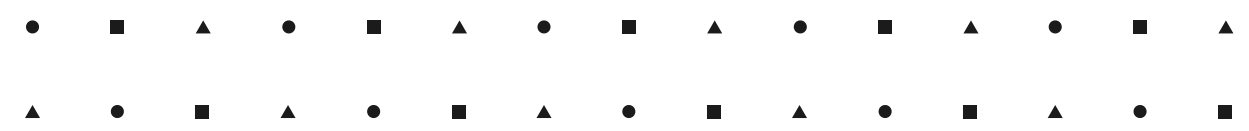




Citas



Inventario



Doctores

Codificacion Crud Citas

```
import android.content.DialogInterface;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import sv.edu.udb.proyectodsm.datos.Citas;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
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.Query;
import com.google.firebase.database.ValueEventListener;

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

public class CitasActivity extends AppCompatActivity{
    public static FirebaseDatabase database = FirebaseDatabase.getInstance();
    public static DatabaseReference refCitas = database.getReference("cita");

    Query consultarOrdenada = refCitas.orderByChild("nombre");

    List<Citas> cita;
    ListView listaCitas;

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

        inicializar();
    }

    private void inicializar() {
        FloatingActionButton fab_agregar= findViewById(R.id.fab_agregar);
        listaCitas = findViewById(R.id.ListaCitas);

        // Cuando el usuario haga clic en la lista (para editar registro)
        listaCitas.setOnItemClickListener(new AdapterView.OnItemClickListener()
        {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                Intent i = new Intent(CitasActivity.this, EditarCitas.class);
                i.putExtra("id", id);
                startActivity(i);
            }
        });
    }
}
```

Codificacion Crud Citas

```
1, long l) {
    Intent intent = new Intent(getApplicationContext(),
AddCitasActivity.class);

    intent.putExtra("accion","e"); // Editar
    intent.putExtra("key", cita.get(i).getKey());
    intent.putExtra("nombre",cita.get(i).getNombre());
    intent.putExtra("dui",cita.get(i).getDui());
    intent.putExtra("fecha",cita.get(i).getFecha_cita());
    intent.putExtra("hora",cita.get(i).getHora());
    intent.putExtra("descripcion",cita.get(i).getDescripcion());
    startActivity(intent);
    }
});

// Cuando el usuario hace un LongClic (clic sin soltar elemento por mas
de 2 segundos)
// Es por que el usuario quiere eliminar el registro
listaCitas.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
    @Override
    public boolean onItemLongClick(AdapterView<?> adapterView, View
view, final int position, long l) {

        // Preparando cuadro de dialogo para preguntar al usuario
        // Si esta seguro de eliminar o no el registro
        AlertDialog.Builder ad = new
AlertDialog.Builder(CitasActivity.this);
        ad.setMessage("Está seguro de eliminar registro?")
            .setTitle("Confirmación");

        ad.setPositiveButton("Si", new DialogInterface.OnClickListener()
{
            public void onClick(DialogInterface dialog, int id) {
                CitasActivity.refCitas

                .child(cita.get(position).getKey()).removeValue();

                Toast.makeText(CitasActivity.this,
                    "Registro borrado!",Toast.LENGTH_SHORT).show();
            }
        });
        ad.setNegativeButton("No", new DialogInterface.OnClickListener()
{
            public void onClick(DialogInterface dialog, int id) {
                Toast.makeText(CitasActivity.this,
                    "Operación de borrado
cancelada!",Toast.LENGTH_SHORT).show();
            }
        });

        ad.show();
        return true;
    }
});
}
```

Codificacion Crud Citas

```
fab_agregar.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        // Cuando el usuario quiere agregar un nuevo registro  
        Intent i = new Intent(getBaseContext(), AddCitasActivity.class);  
        i.putExtra("accion","a"); // Agregar  
        i.putExtra("key","");  
        i.putExtra("nombre","");  
        i.putExtra("dui","");  
        i.putExtra("fecha","");  
        i.putExtra("hora","");  
        i.putExtra("descripcion","");  
        startActivity(i);  
    }  
});  
  
cita = new ArrayList<>();  
  
// Cambiarlo refProductos a consultaOrdenada para ordenar lista  
consultarOrdenada.addValueEventListener(new ValueEventListener() {  
    @Override  
    public void onDataChange(DataSnapshot dataSnapshot) {  
        // Procedimiento que se ejecuta cuando hubo algun cambio  
        // en la base de datos  
        // Se actualiza la coleccion de personas  
        cita.removeAll(cita);  
        for (DataSnapshot dato : dataSnapshot.getChildren()) {  
            Citas citas = dato.getValue(Citas.class);  
            citas.setKey(dato.getKey());  
            cita.add(citas);  
        }  
  
        AdaptadorCitas adapter = new AdaptadorCitas(CitasActivity.this,  
            cita );  
        listaCitas.setAdapter(adapter);  
    }  
    @Override  
    public void onCancelled(DatabaseError databaseError) {  
    }  
});  
}
```

Codificacion Crud Citas

```
fab_agregar.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        // Cuando el usuario quiere agregar un nuevo registro  
        Intent i = new Intent(getBaseContext(), AddCitasActivity.class);  
        i.putExtra("accion","a"); // Agregar  
        i.putExtra("key","");  
        i.putExtra("nombre","");  
        i.putExtra("dui","");  
        i.putExtra("fecha","");  
        i.putExtra("hora","");  
        i.putExtra("descripcion","");  
        startActivity(i);  
    }  
});  
  
cita = new ArrayList<>();  
  
// Cambiarlo refProductos a consultaOrdenada para ordenar lista  
consultarOrdenada.addValueEventListener(new ValueEventListener() {  
    @Override  
    public void onDataChange(DataSnapshot dataSnapshot) {  
        // Procedimiento que se ejecuta cuando hubo algun cambio  
        // en la base de datos  
        // Se actualiza la coleccion de personas  
        cita.removeAll(cita);  
        for (DataSnapshot dato : dataSnapshot.getChildren()) {  
            Citas citas = dato.getValue(Citas.class);  
            citas.setKey(dato.getKey());  
            cita.add(citas);  
        }  
  
        AdaptadorCitas adapter = new AdaptadorCitas(CitasActivity.this,  
            cita );  
        listaCitas.setAdapter(adapter);  
    }  
    @Override  
    public void onCancelled(DatabaseError databaseError) {  
    }  
});  
}
```

Codificacion Crud Citas

```
import android.os.Bundle;
import android.os.PersistableBundle;
import android.view.View;
import android.widget.EditText;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import sv.edu.udb.proyectodsm.datos.Citas;

import android.app.DatePickerDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import java.util.Calendar;
import android.app.TimePickerDialog;
import android.widget.TimePicker;

public class AddCitasActivity extends AppCompatActivity{
    EditText edtDUI,edtNombre,edtFecha,edtHora,edtDescripcion;
    String key="",nombre="",dui="",fecha="",hora="",descripcion="",accion="";
    Button buttonSave;
    int cyear, cday, cmonth;
    int chour,cminute;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_citas);
        inicializar();
    }
    private void inicializar() {
        edtNombre = findViewById(R.id.edtNombre);
        edtDUI = findViewById(R.id.edtDUI);
        edtFecha = findViewById(R.id.edtFech);
        edtHora = findViewById(R.id.edtHora);
        edtDescripcion = findViewById(R.id.edtDescripcion);

        edtFecha.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                final Calendar calendar=Calendar.getInstance();
                cyear = calendar.get(Calendar.YEAR);
                cmonth = calendar.get(Calendar.MONTH);
                cday = calendar.get(Calendar.DAY_OF_MONTH);

                DatePickerDialog datePickerDialog = new
                DatePickerDialog(AddCitasActivity.this, new DatePickerDialog.OnDateSetListener()
                {
```


Codificacion ADD CITAS

```
import android.os.Bundle;
import android.os.PersistableBundle;
import android.view.View;
import android.widget.EditText;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import sv.edu.udb.proyectodsm.datos.Citas;

import android.app.DatePickerDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import java.util.Calendar;
import android.app.TimePickerDialog;
import android.widget.TimePicker;

public class AddCitasActivity extends AppCompatActivity{
    EditText edtDUI,edtNombre,edtFecha,edtHora,edtDescripcion;
    String key="",nombre="",dui="",fecha="",hora="",descripcion="",accion="";
    Button buttonSave;
    int cyear, cday, cmonth;
    int chour,cminute;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_citas);
        inicializar();
    }
    private void inicializar() {
        edtNombre = findViewById(R.id.edtNombre);
        edtDUI = findViewById(R.id.edtDUI);
        edtFecha = findViewById(R.id.edtFech);
        edtHora = findViewById(R.id.edtHora);
        edtDescripcion = findViewById(R.id.edtDescripcion);

        edtFecha.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                final Calendar calendar=Calendar.getInstance();
                cyear = calendar.get(Calendar.YEAR);
                cmonth = calendar.get(Calendar.MONTH);
                cday = calendar.get(Calendar.DAY_OF_MONTH);

                DatePickerDialog datePickerDialog = new
                DatePickerDialog(AddCitasActivity.this, new DatePickerDialog.OnDateSetListener()
                {
```


Codificacion ADD CITAS

```
public void onDateSet(DatePicker view, int year, int month,
int dayOfMonth) {
    edtFecha.setText(dayOfMonth+"/"+(month+1)+"/"+year);
}
},cyear,cmonth,cday);
datePickerDialog.show();
});

edtHora.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        final Calendar calendar = Calendar.getInstance();
        chour=calendar.get(Calendar.HOUR_OF_DAY);
        cminute=calendar.get(Calendar.MINUTE);

        TimePickerDialog timePickerDialog = new
TimePickerDialog(AddCitasActivity.this, new TimePickerDialog.OnTimeSetListener()
{
    @Override
    public void onTimeSet(TimePicker view, int hourOfDay, int
minute) {
        edtHora.setText(hourOfDay+":"+minute);
    }
},chour,cminute,false);
timePickerDialog.show();
});

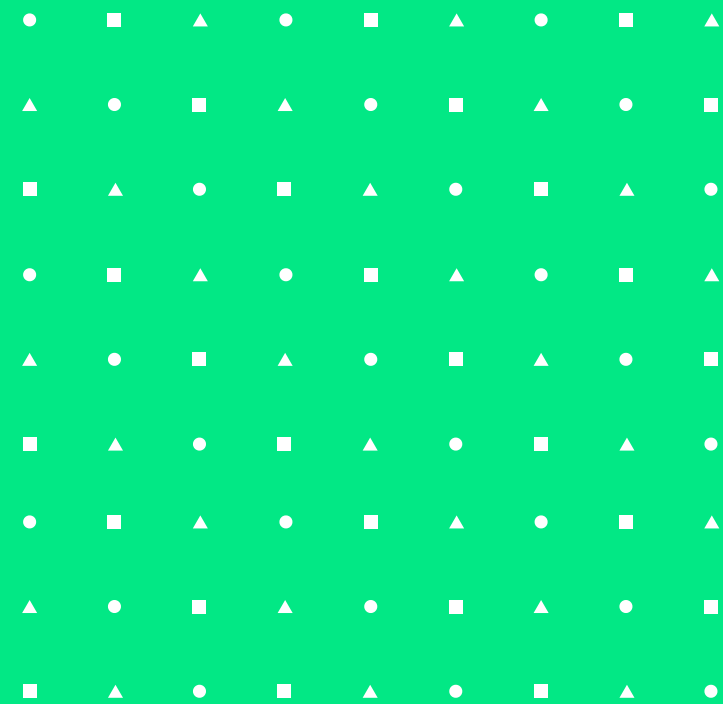
// Obtención de datos que envia actividad anterior
Bundle datos = getIntent().getExtras();
key = datos.getString("key");
dui = datos.getString("dui");
nombre=datos.getString("nombre");
fecha=datos.getString("fecha");
hora=datos.getString("hora");
descripcion=datos.getString("descripcion");

accion=datos.getString("accion");
edtDUI.setText(dui);
edtNombre.setText(nombre);
edtFecha.setText(fecha);
edtHora.setText(hora);
edtDescripcion.setText(descripcion);
}

public void guardar(View v){
    String nombre = edtNombre.getText().toString();
    String dui = edtDUI.getText().toString();
    String fecha = edtFecha.getText().toString();
    String hora = edtHora.getText().toString();
    String descripcion = edtDescripcion.getText().toString();
    // Se forma objeto persona
    Citas cita = new Citas(dui,nombre,fecha,hora,descripcion);
```

Codificacion ADD CITAS

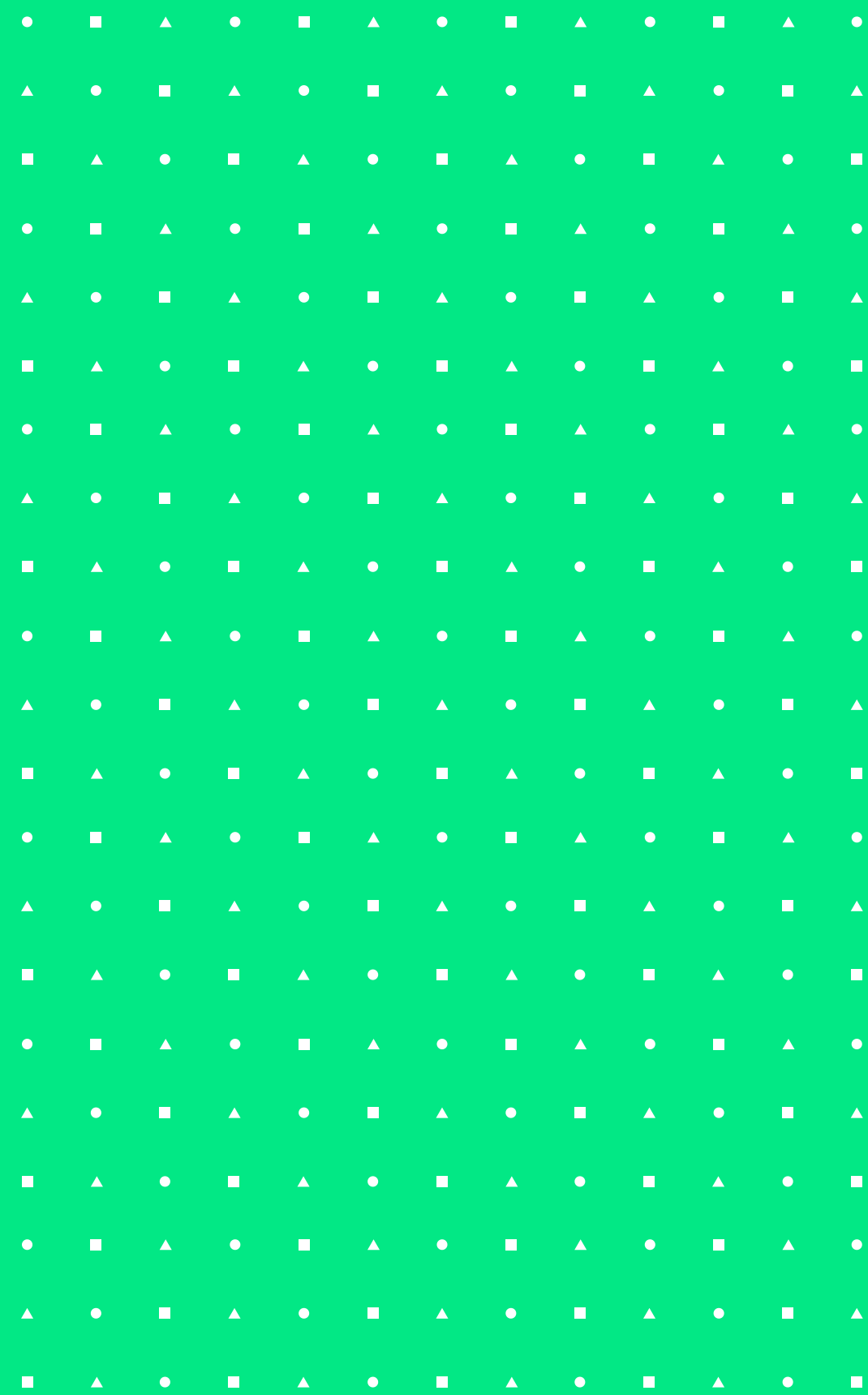
```
        if (accion.equals("a")) { //Agregar usando push()
            CitasActivity.refCitas.push().setValue(cita);
        }
        else // Editar usando setValue
        {
            CitasActivity.refCitas.child(key).setValue(cita);
        }
        finish();
    }
    public void cancelar(View v){
        finish();
    }
}
```



Finalizando

En este manual técnico se describen los componentes básicos, todo esto con el objetivo de que se pueda leer, interpretar y analizar las partes de que se conforma el sistema desarrollado a fin de que se le desee realizar modificaciones futuras o bien actualizaciones para mejorar su eficiencia y de ser posible sea base para algunos sistemas futuros a desarrollarse que sean afines a este.





K/B

¡Gracias!

Si lo deseas, puedes preguntarnos lo que quieras.