

Universidad Tecnológica de El Salvador

FACULTAD DE INFORMÁTICA Y CIENCIAS APLICADAS

ESCUELA DE INFORMÁTICA



ASIGNATURA: Electiva 1

SECCION:01

Documento:

Ejercicio para evaluación 4

TRABAJO PRESENTADO POR:

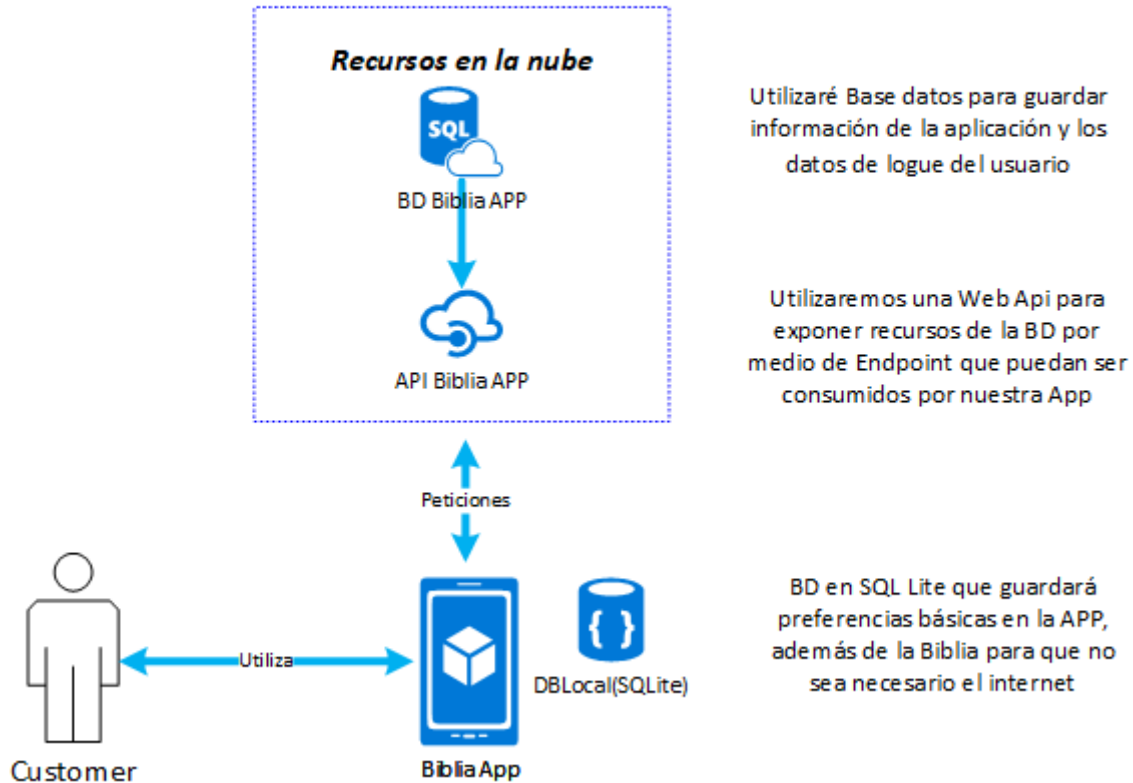
INTEGRANTES

NOMBRES	CARNET
Perdomo Aragón, Roberto Esaú	29-4645-2013

22 de diciembre de 2020de 2020

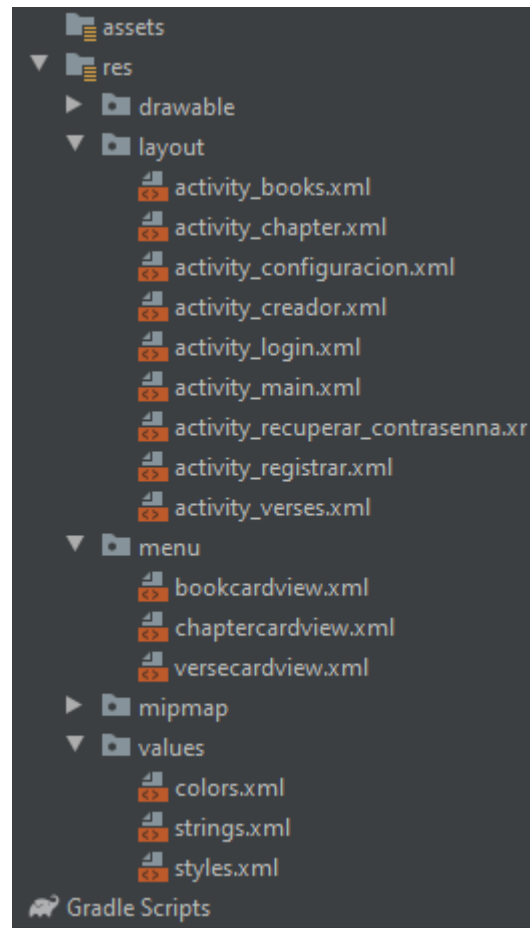
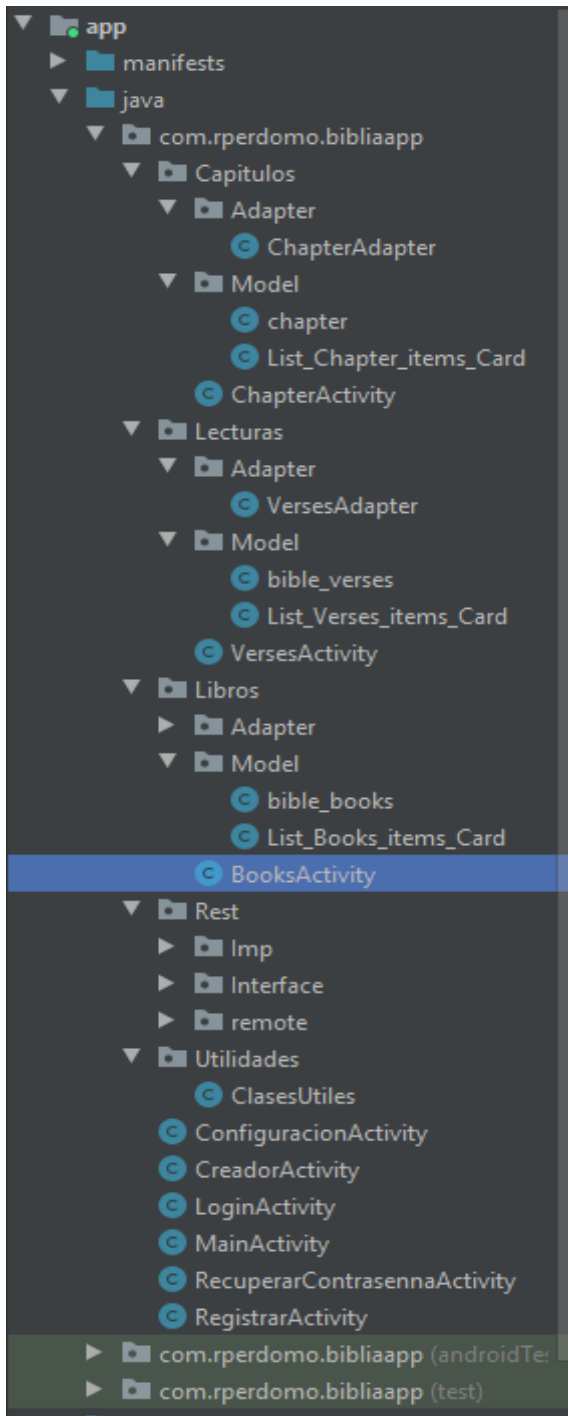
Arquitectura del proyecto

Infraestructura para Biblia App movil



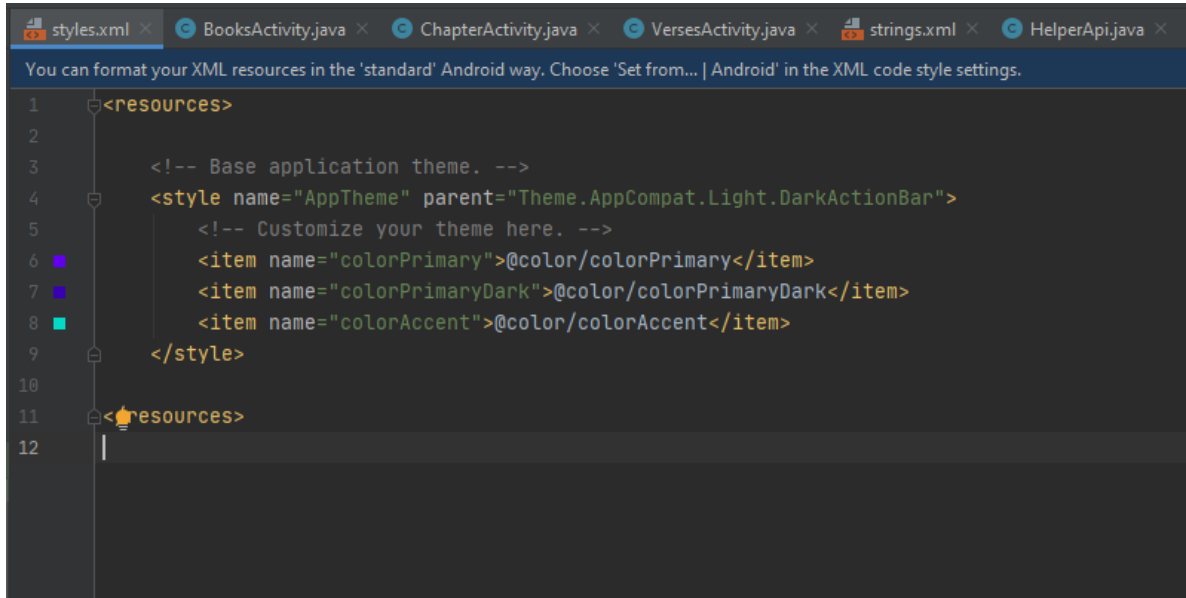
Capturas de Código

árbol del proyecto



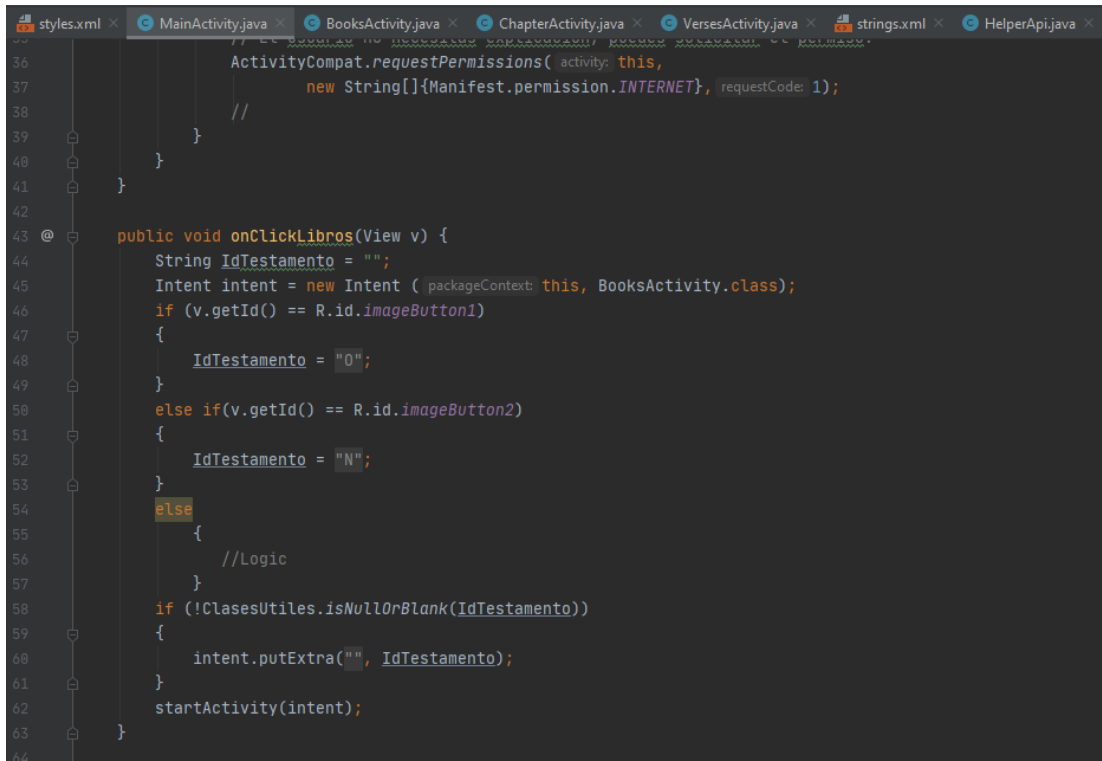
Constantes utilizadas en la aplicación y su uso con los intentos

Los utilizamos para nombrar los intent de forma statica y también para mostrar unos textos en las vistas

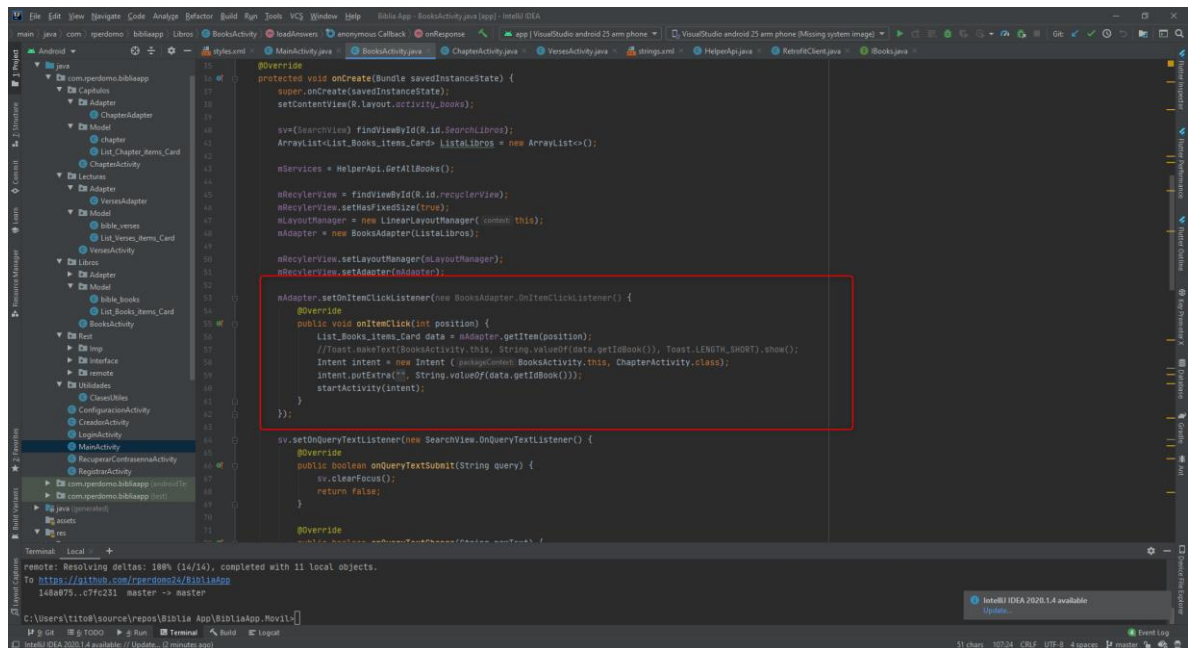


```
1 <resources>
2
3 <!-- Base application theme. -->
4 <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
5 <!-- Customize your theme here. -->
6 <item name="colorPrimary">@color/colorPrimary</item>
7 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
8 <item name="colorAccent">@color/colorAccent</item>
9 </style>
10
11 <!--resources>
12
```

utilización de Intent en la aplicación



```
36 // << REMOVED --> << REMOVED --> << REMOVED --> << REMOVED --> << REMOVED --> << REMOVED -->
37 ActivityCompat.requestPermissions( activity: this,
38 new String[]{Manifest.permission.INTERNET}, requestCode: 1);
39 //
40 }
41 }
42
43 @
44 public void onClickLibros(View v) {
45 String IdTestamento = "";
46 Intent intent = new Intent ( packageContext: this, BooksActivity.class);
47 if (v.getId() == R.id.imageButton1)
48 {
49 IdTestamento = "0";
50 }
51 else if(v.getId() == R.id.imageButton2)
52 {
53 IdTestamento = "N";
54 }
55 else
56 {
57 //Logic
58 }
59 if (!ClasesUtiles.isNullOrBlank(IdTestamento))
60 {
61 intent.putExtra("", IdTestamento);
62 }
63 startActivity(intent);
64 }
```



```

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

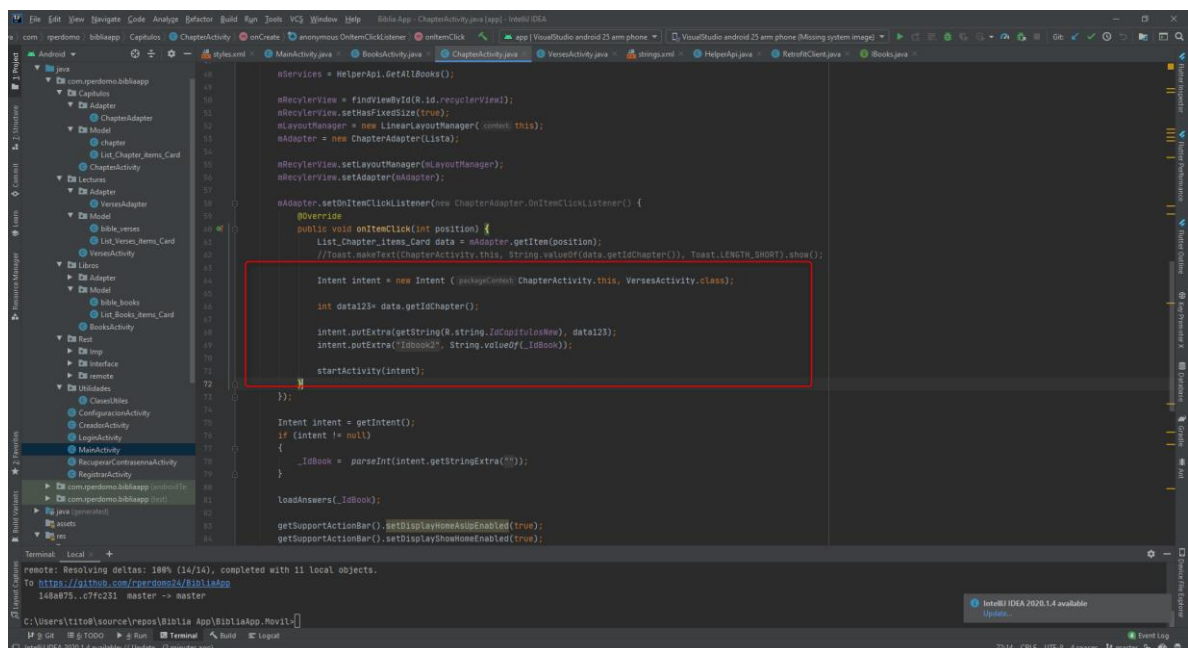
    sv(SearchView) findViewById(R.id.searchLibros);
    ArrayList<List_Books_Items_Card> listaLibros = new ArrayList<>();
    nServices = HelperApi.getAllBooks();

    nRecyclerView = findViewById(R.id.recyclerView);
    nRecyclerView.setHasFixedSize(true);
    nLayoutManager = new LinearLayoutManager(this);
    nAdapter = new BooksAdapter(listaLibros);
    nRecyclerView.setLayoutManager(nLayoutManager);
    nRecyclerView.setAdapter(nAdapter);

    nAdapter.setOnItemClickListener(new BooksAdapter.OnItemClickListener() {
        @Override
        public void onItemClick(int position) {
            List_Books_Items_Card data = nAdapter.getItem(position);
            //Toast.makeText(this, String.valueOf(data.getIdBook()), Toast.LENGTH_SHORT).show();
            Intent intent = new Intent(getApplicationContext(), ChapterActivity.class);
            intent.putExtra("id", String.valueOf(data.getIdBook()));
            startActivity(intent);
        }
    });

    sv.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String query) {
            sv.clearFocus();
            return false;
        }

        @Override
        public boolean onQueryTextChange(String query) {
            return false;
        }
    });
}
    
```



```

nServices = HelperApi.getAllBooks();
nRecyclerView = findViewById(R.id.recyclerView);
nRecyclerView.setHasFixedSize(true);
nLayoutManager = new LinearLayoutManager(this);
nAdapter = new ChapterAdapter(lista);
nRecyclerView.setLayoutManager(nLayoutManager);
nRecyclerView.setAdapter(nAdapter);

nAdapter.setOnItemClickListener(new ChapterAdapter.OnItemClickListener() {
    @Override
    public void onItemClick(int position) {
        List_Chapter_Items_Card data = nAdapter.getItem(position);
        //Toast.makeText(this, String.valueOf(data.getIdChapter()), Toast.LENGTH_SHORT).show();

        Intent intent = new Intent(getApplicationContext(), VersesActivity.class);
        int dataId23 = data.getIdChapter();
        intent.putExtra(getString(R.string.idChapter), dataId23);
        intent.putExtra("idBook", String.valueOf(data.getIdBook()));
        startActivity(intent);
    }
});

Intent intent = getIntent();
if (intent != null) {
    _idBook = parseInt(intent.getStringExtra("id"));
}

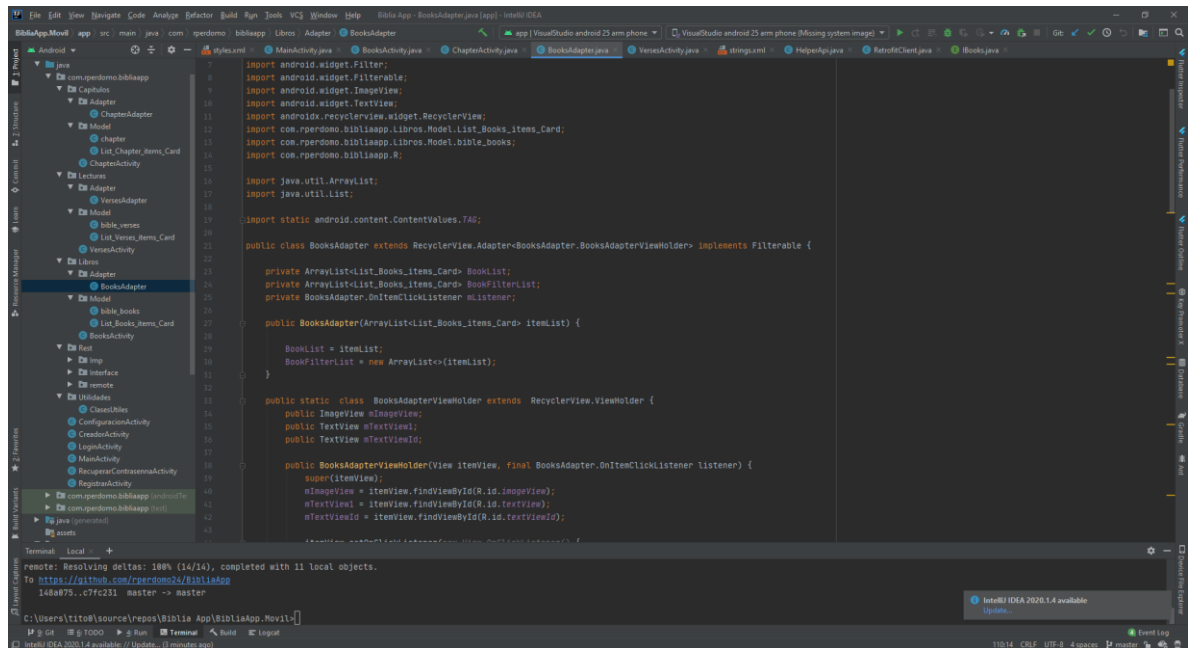
loadAnswers(_idBook);
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
}
    
```

```
Intent intent = getIntent();
if (intent != null)
{
    _IdBook = parseInt(intent.getStringExtra(""));
}

loadAnswers(_IdBook);
```

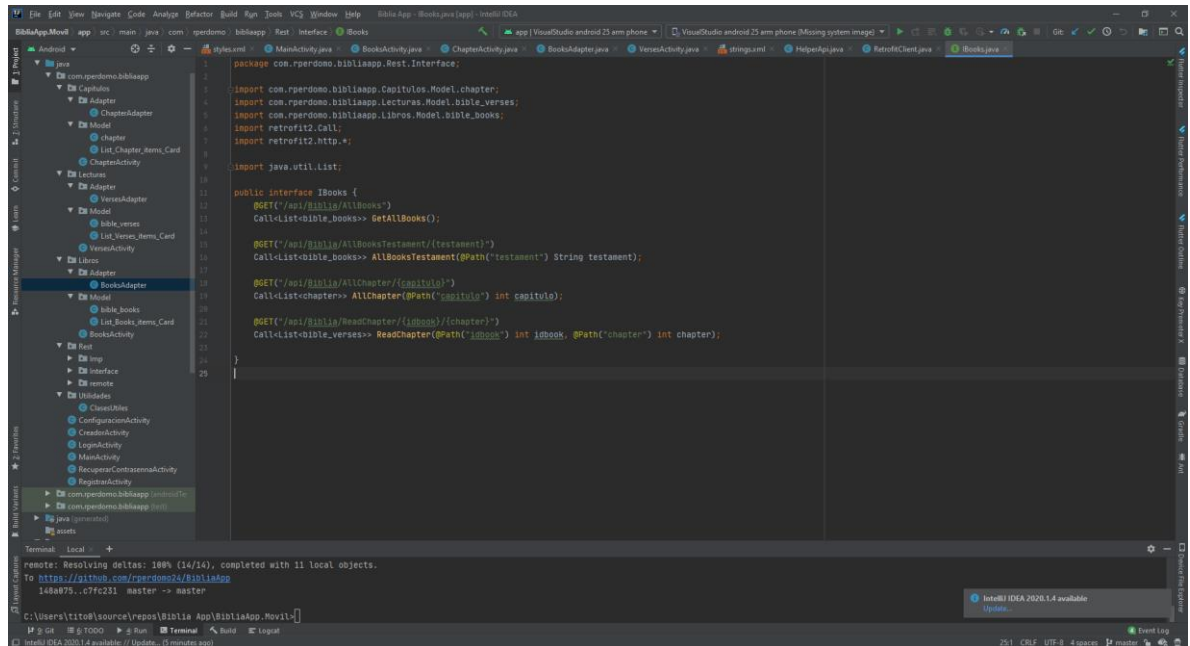
Manejo de Listas

Manejo de listas por medio de Recycle View y utilizamos Adapter para el manejo dinámico de listas



Consumo de Api Rest

Se utiliza la librería retrofit2.Call y también com.squareup.retrofit2:converter-gson:2.9.0 para convertir a JSON



```
package com.rperdomo.bibliaapp.Rest.Interface;

import com.rperdomo.bibliaapp.Capitulos.Model.chapter;
import com.rperdomo.bibliaapp.Lecturas.Model.bible_verses;
import com.rperdomo.bibliaapp.Libros.Model.bible_books;
import retrofit2.Call;
import retrofit2.http.*;

import java.util.List;

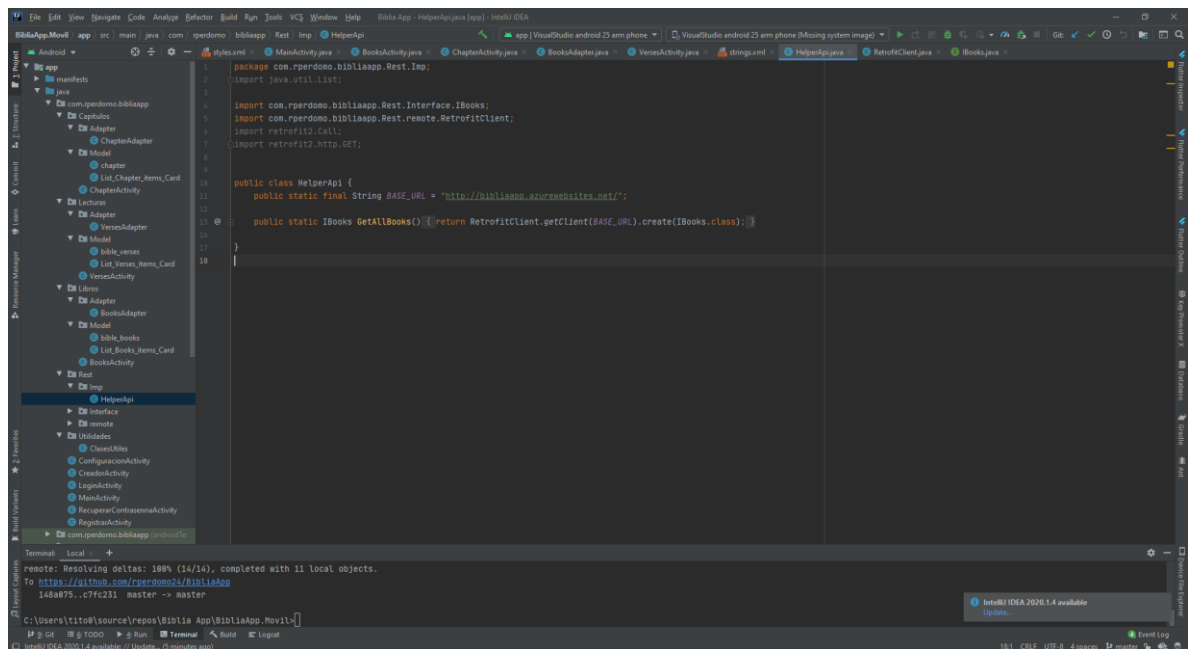
public interface IBooks {

    @GET("/api/biblia/AllBooks")
    Call<List<bible_books>> GetAllBooks();

    @GET("/api/biblia/AllBooksTestament/{testament}")
    Call<List<bible_books>> AllBooksTestament(@Path("testament") String testament);

    @GET("/api/biblia/AllChapter/{capitulo}")
    Call<List<chapter>> AllChapter(@Path("capitulo") int capitulo);

    @GET("/api/biblia/ReadChapter/{idbook}/{chapter}")
    Call<List<bible_verses>> ReadChapter(@Path("idbook") int idbook, @Path("chapter") int chapter);
}
```



```
package com.rperdomo.bibliaapp.Rest.Imp;

import java.util.List;

import com.rperdomo.bibliaapp.Rest.Interface.IBooks;
import com.rperdomo.bibliaapp.Rest.remote.RetrofitClient;
import retrofit2.Call;
import retrofit2.http.GET;

public class HelperApi {

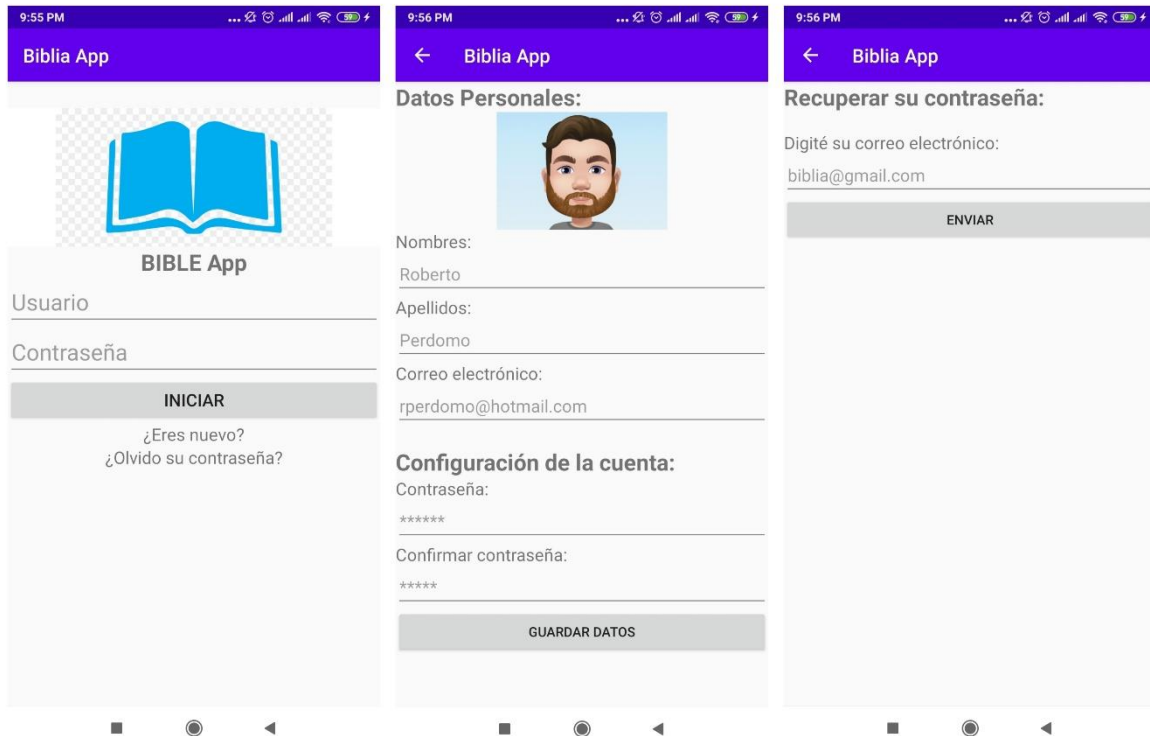
    public static final String BASE_URL = "http://bibliamundo.org/webservices/get/";

    public static IBooks GetAllBooks() {return RetrofitClient.getClient(BASE_URL).create(IBooks.class); }
}
```

Capturas aplicación en funcionamiento

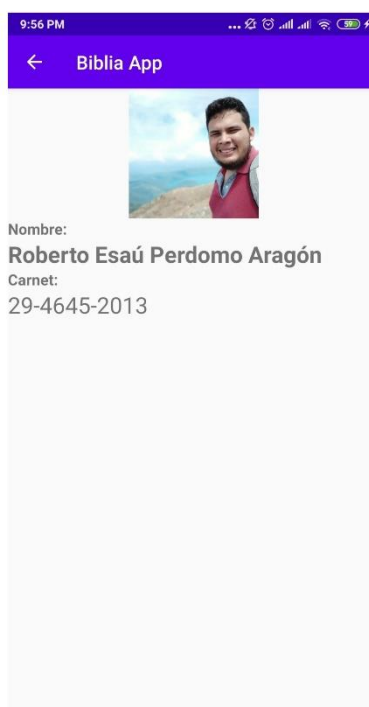
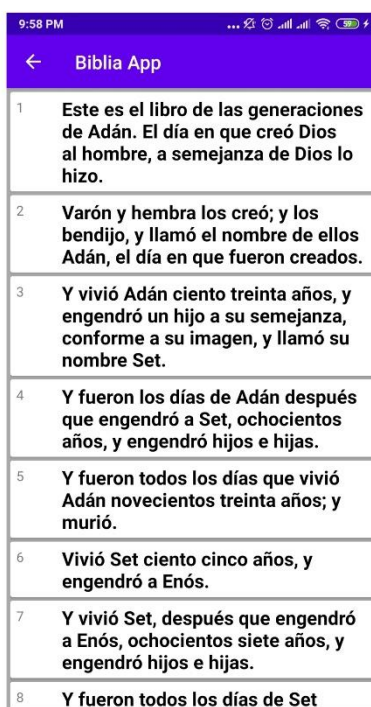
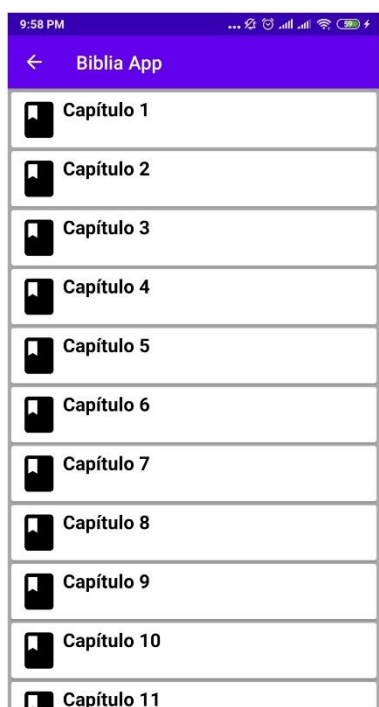
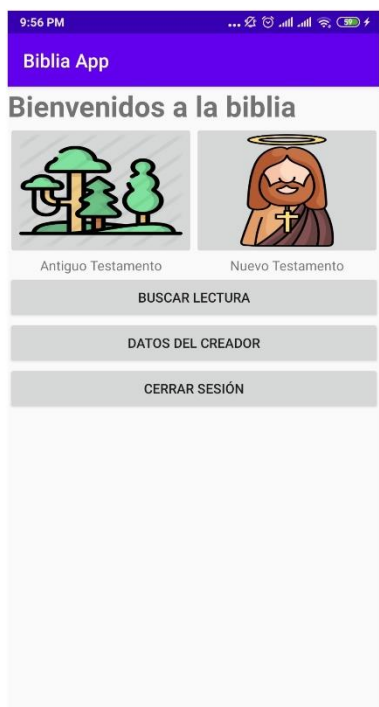
Capturas en general de todo el flujo.

Se creo un login pero por temas de tiempo laboral no esta funcional (No hay logeo de usuario solo el mock)



LINK DE VIDEO DE LA APLICACIÓN FUNCIONANDO

<https://youtu.be/vI5l6Yxacn4>



Consultas a la API

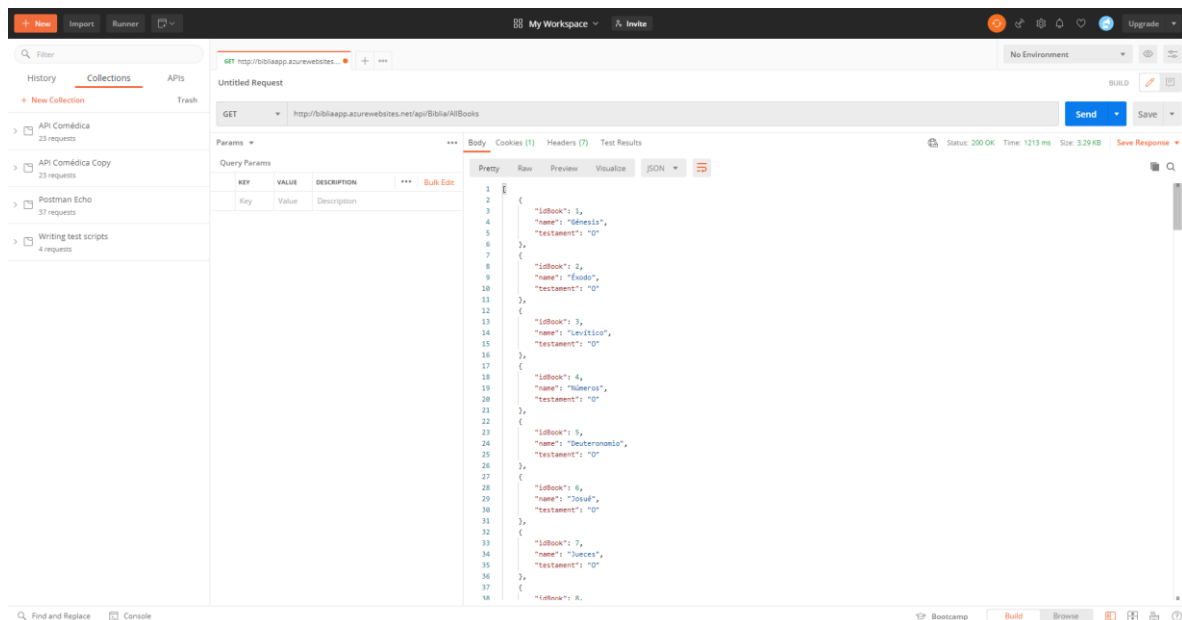
```
Select C:\Users\tito\Documents\ngrok\ngrok.exe
ngrok by @inconsheveable

Session Status      online
Account             Robert Perdomo (Plan: Free)
Version             2.3.35
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://30d3391d2a50.ngrok.io -> https://localhost:44382
Forwarding           https://30d3391d2a50.ngrok.io -> https://localhost:44382

Connections         ttl    opn    rt1    rt5    p50    p90
                   1      1      0.00   0.00   77.30  77.30

HTTP Requests
-----
GET /api/Biblia/ReadChapter/5/9      200 OK
GET /api/Biblia/AllChapter/5        200 OK
GET /api/Biblia/AllBooks            200 OK
GET /api/Biblia/AllBooksTestament/O 200 OK
GET /api/Biblia/AllBooksTestament/N 200 OK
GET /api/Biblia/AllBooksTestament/O 200 OK
GET /api/Biblia/AllBooksTestament/O 200 OK
GET /api/Biblia/ReadChapter/1/10    200 OK
GET /api/Biblia/ReadChapter/1/6     200 OK
GET /api/Biblia/AllChapter/1        200 OK
```

Respuestas de API



Bibliografía

Video de aplicación en funcionamiento enlace a YouTube

<https://youtu.be/vl5l6Yxacn4>

Código Fuente creado por Estudiante

<https://github.com/rperdomo24/BibliaApp>

URL API

<http://bibliaapp.azurewebsites.net/>