



Tecnológico Nacional de
México

Instituto Tecnológico de
Reynosa

Materia: Interacción humano computadora
y Desarrollo de aplicaciones móviles

Maestro: Dora Luz Quevedo Valenzuela

Título: Aplicación en duplas (Traductor de
señas a texto y texto a señas)

Fecha de entrega: 24/10/22

Equipo:

Castillo Jr. Gregorio 19580589

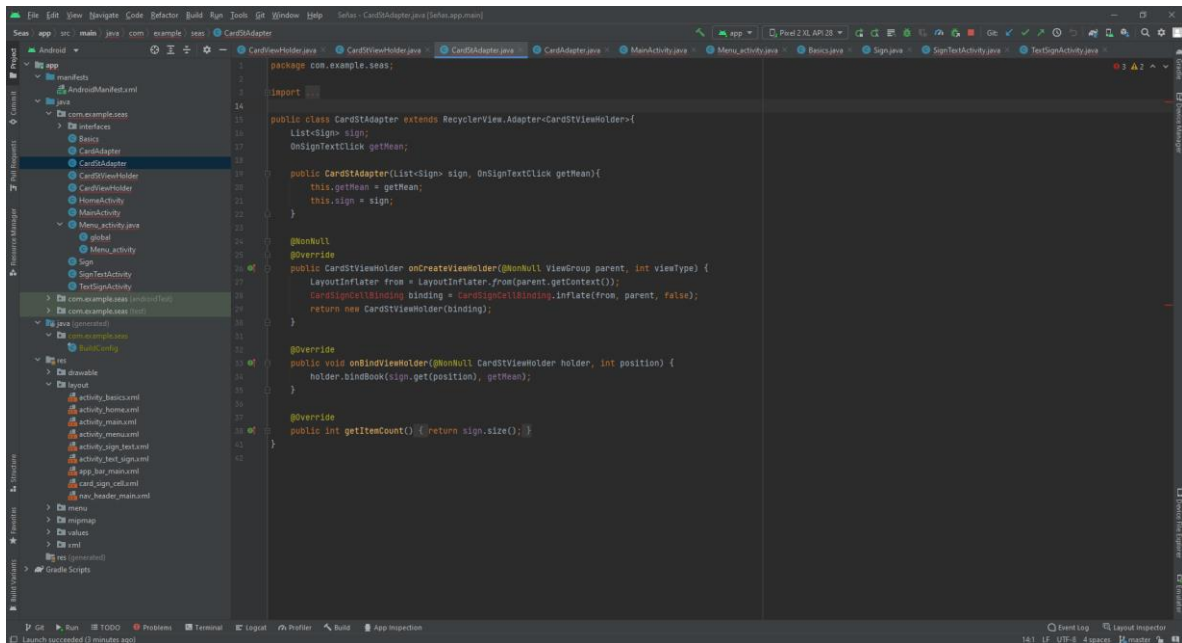
Hernández Moreno Eduardo 19580609

Índice

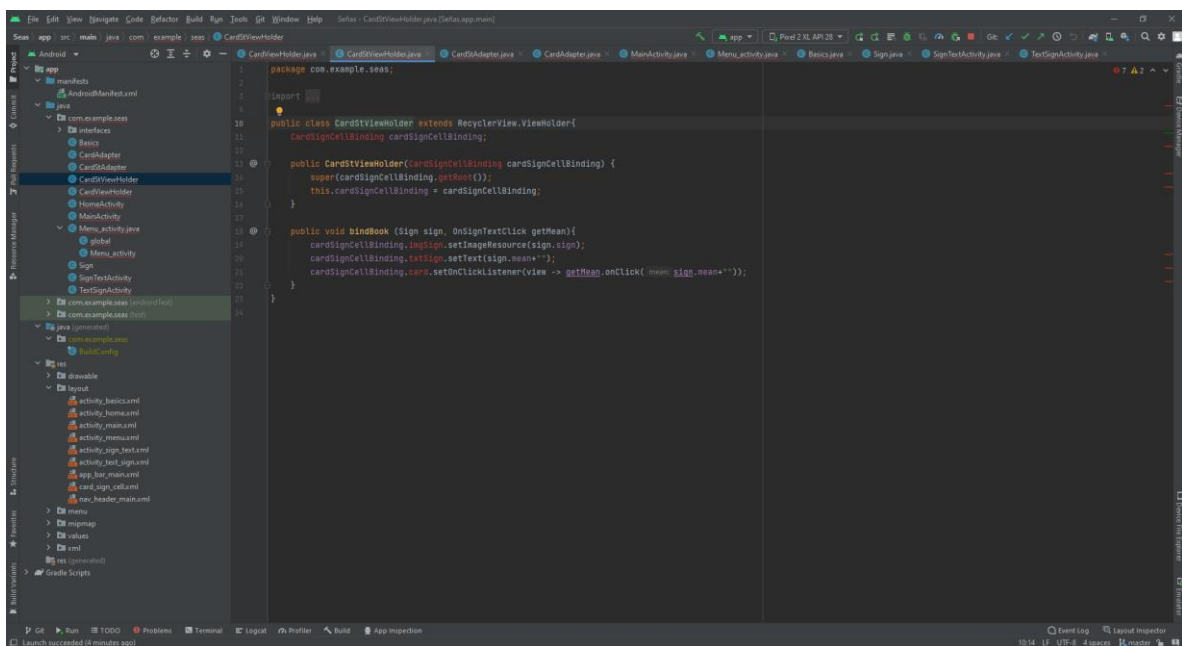
Índice	1
Imágenes de la codificación.....	2
Explicación de la codificación	12
Pagina principal	12
Texto a seña	12
Seña a texto	13
Menú de señas básicas	13
Señas básicas	14
Imágenes de la ejecución	15
Resultados	23



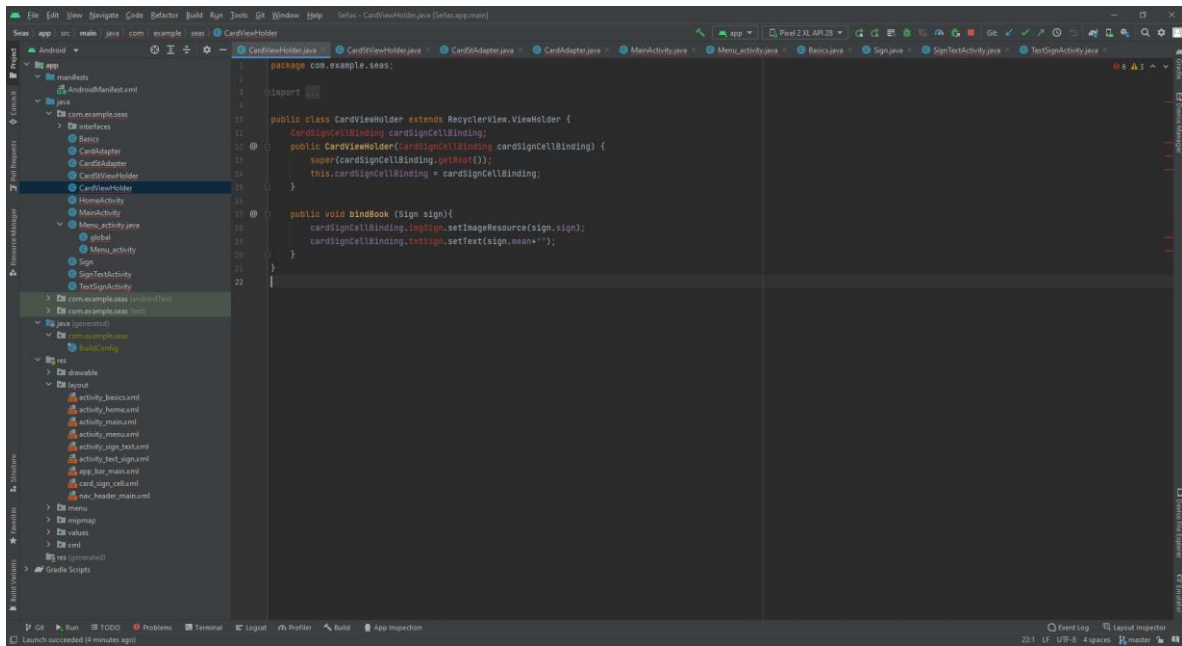
CardStAdapter



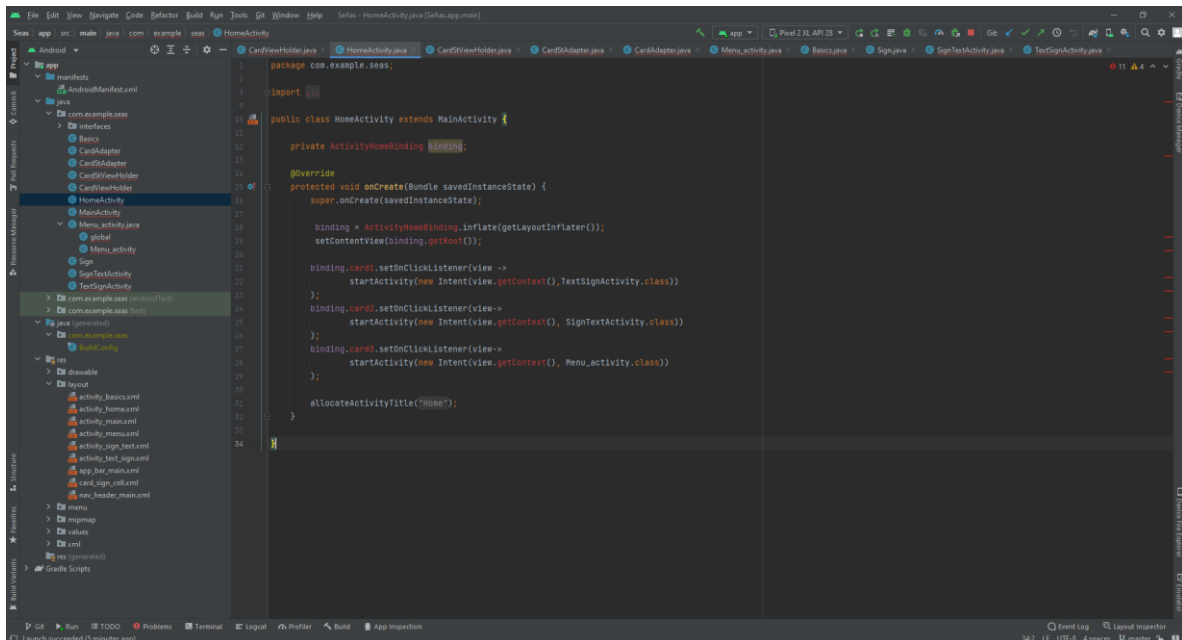
CardViewHolder



CardViewHolder



HomeActivity



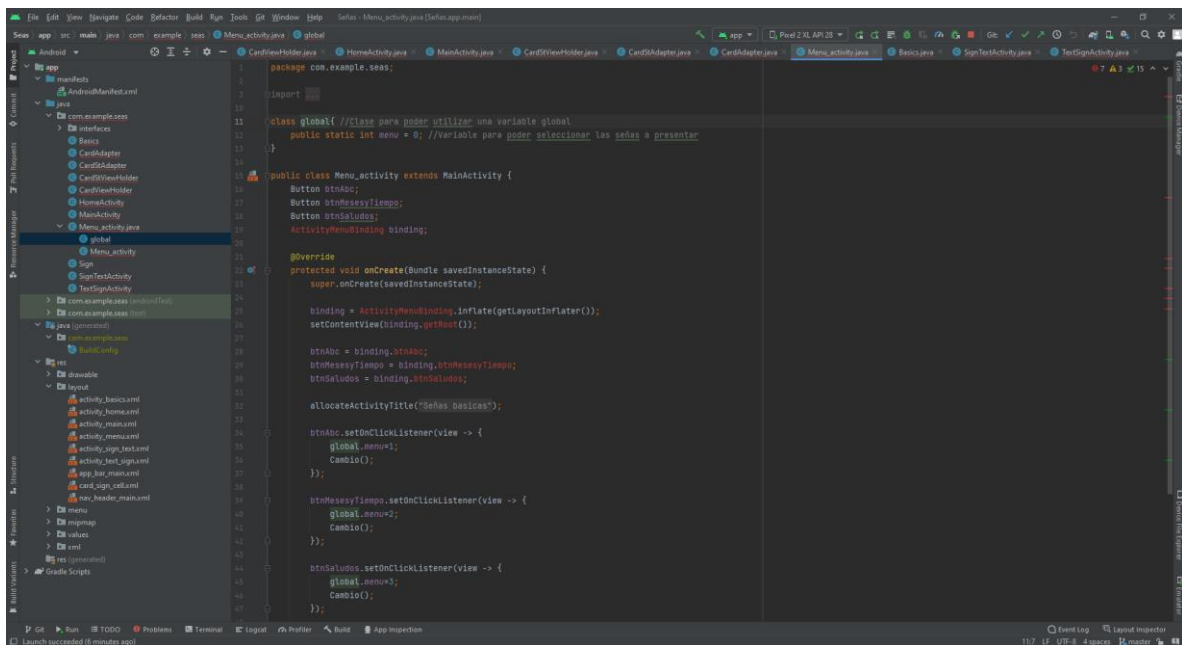
MainActivity

```

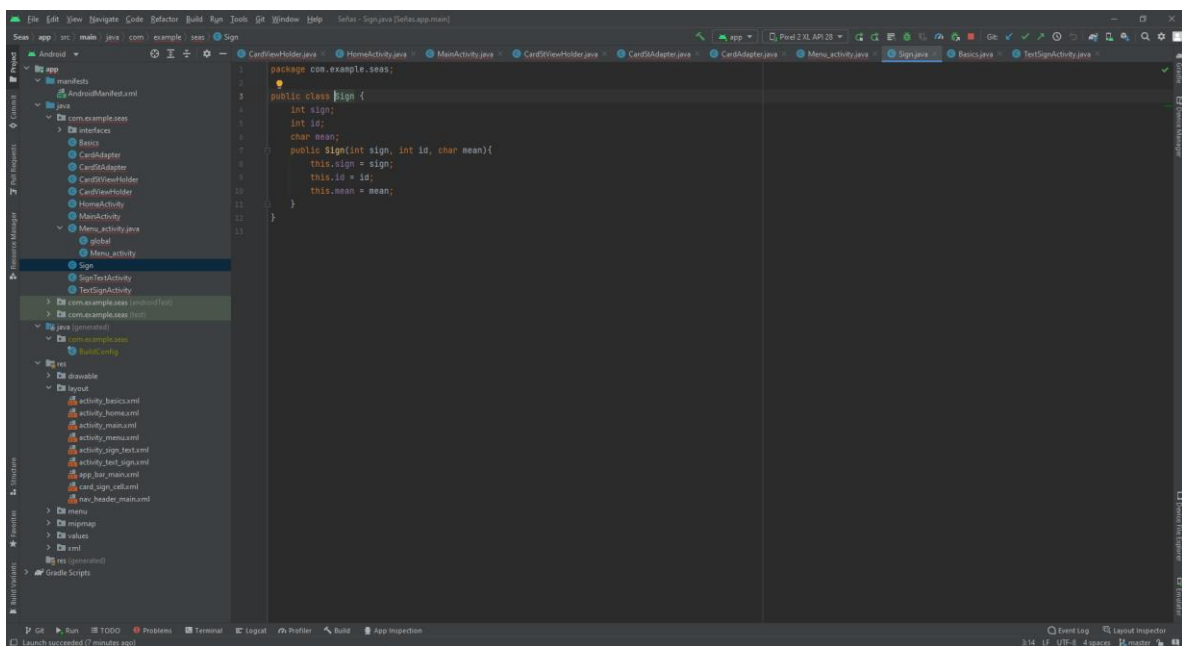
1  package com.example.sas;
2
3  import androidx.appcompat.app.AppCompatActivity;
4  import androidx.navigation.NavController;
5  import androidx.navigation.Navigation;
6  import androidx.navigation.ui.AppBarConfiguration;
7  import androidx.navigation.ui.NavigationUI;
8  import androidx.drawerlayout.widget.DrawerLayout;
9  import androidx.appcompat.app.AppCompatActivity;
10 import androidx.appcompat.widget.Toolbar;
11
12 import android.os.Bundle;
13 import android.view.MenuItem;
14
15 public class MainActivity extends AppCompatActivity {
16
17     private AppBarConfiguration mAppBarConfiguration;
18     private DrawerLayout drawer;
19     private NavController navController;
20     private Toolbar toolbar;
21
22     @Override
23     protected void onCreate(Bundle savedInstanceState) {
24         super.onCreate(savedInstanceState);
25         setContentView(R.layout.activity_main);
26         drawer = findViewById(R.id.drawer_layout);
27         navController = Navigation.findNavController(this);
28         mAppBarConfiguration = new AppBarConfiguration.Builder(
29             R.id.nav_home, R.id.nav_sign, R.id.nav_basic_sign)
30             .setDrawerLayout(drawer).build();
31         navController.addOnItemSelectedListener(selectedItemClicked);
32         NavigationUI.setupActionBarWithNavController(this, navController, mAppBarConfiguration);
33     }
34
35     @Override
36     public boolean onOptionsItemSelected() {
37         return super.onOptionsItemSelected();
38     }
39
40     private void selectedItemClicked() {
41         NavController navController = Navigation.findNavController(this);
42         DrawerLayout drawer = findViewById(R.id.drawer_layout);
43         switch (navController.getCurrentDestination().getId()) {
44             case R.id.nav_home:
45                 startActivity(new Intent(this, HomeActivity.class));
46                 overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
47                 break;
48             case R.id.nav_sign:
49                 startActivity(new Intent(this, SignTextActivity.class));
50                 overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
51                 break;
52             case R.id.nav_basic_sign:
53                 startActivity(new Intent(this, SignTextActivity.class));
54                 overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
55                 break;
56             case R.id.nav_basic_sign:
57                 startActivity(new Intent(this, MenuActivity.class));
58                 overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
59                 break;
60         }
61         drawer.closeDrawer(GravityCompat.START);
62     }
63
64     protected void allocateActivityTitle(String titleString) {
65         if (getSupportActionBar() != null) {
66             getSupportActionBar().setTitle(titleString);
67         }
68     }
69 }

```

global



Sign



SignTextActivity

```
package com.example.seas;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import androidx.appcompat.widget.Toolbar;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class SignTextActivity extends AppCompatActivity implements OnSignTextClick {
    List<Sign> signList = new ArrayList<>();
    ActivitySignTextBinding binding;
    TextView textViewResult;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivitySignTextBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        allocateActivityTitle("Seas & Texts");
        setDataSigns();

        RecyclerView recyclerView = binding.recyclerView;
        recyclerView.setLayoutManager(new LinearLayoutManager(getApplicationContext(), LinearLayoutManager.VERTICAL, false));
        recyclerView.setAdapter(new CardListAdapter(signList, getMainTitle()));

        //Listen audio
    }

    private void setDataSigns () {
        signList.add(new Sign(R.drawable.img_a, signList.size(), "mean: 'A'"));
        signList.add(new Sign(R.drawable.img_b, signList.size(), "mean: 'B'"));
        signList.add(new Sign(R.drawable.img_c, signList.size(), "mean: 'C'"));
        signList.add(new Sign(R.drawable.img_d, signList.size(), "mean: 'D'"));
        signList.add(new Sign(R.drawable.img_e, signList.size(), "mean: 'E'"));
        signList.add(new Sign(R.drawable.img_f, signList.size(), "mean: 'F'"));
        signList.add(new Sign(R.drawable.img_g, signList.size(), "mean: 'G'"));
        signList.add(new Sign(R.drawable.img_h, signList.size(), "mean: 'H'"));
        signList.add(new Sign(R.drawable.img_i, signList.size(), "mean: 'I'"));
        signList.add(new Sign(R.drawable.img_j, signList.size(), "mean: 'J'"));
        signList.add(new Sign(R.drawable.img_k, signList.size(), "mean: 'K'"));
        signList.add(new Sign(R.drawable.img_l, signList.size(), "mean: 'L'"));
        signList.add(new Sign(R.drawable.img_m, signList.size(), "mean: 'M'"));
        signList.add(new Sign(R.drawable.img_n, signList.size(), "mean: 'N'"));
        signList.add(new Sign(R.drawable.img_o, signList.size(), "mean: 'O'"));
        signList.add(new Sign(R.drawable.img_p, signList.size(), "mean: 'P'"));
        signList.add(new Sign(R.drawable.img_q, signList.size(), "mean: 'Q'"));
        signList.add(new Sign(R.drawable.img_r, signList.size(), "mean: 'R'"));
        signList.add(new Sign(R.drawable.img_s, signList.size(), "mean: 'S'"));
        signList.add(new Sign(R.drawable.img_t, signList.size(), "mean: 'T'"));
        signList.add(new Sign(R.drawable.img_u, signList.size(), "mean: 'U'"));
        signList.add(new Sign(R.drawable.img_v, signList.size(), "mean: 'V'"));
        signList.add(new Sign(R.drawable.img_w, signList.size(), "mean: 'W'"));
        signList.add(new Sign(R.drawable.img_x, signList.size(), "mean: 'X'"));
        signList.add(new Sign(R.drawable.img_y, signList.size(), "mean: 'Y'"));
        signList.add(new Sign(R.drawable.img_z, signList.size(), "mean: 'Z'"));
        signList.add(new Sign(R.drawable.img_blank, 0, "mean: ''"));
    }
}
```

```
private void setDataSigns () {
    signList.add(new Sign(R.drawable.img_a, signList.size(), "mean: 'A'"));
    signList.add(new Sign(R.drawable.img_b, signList.size(), "mean: 'B'"));
    signList.add(new Sign(R.drawable.img_c, signList.size(), "mean: 'C'"));
    signList.add(new Sign(R.drawable.img_d, signList.size(), "mean: 'D'"));
    signList.add(new Sign(R.drawable.img_e, signList.size(), "mean: 'E'"));
    signList.add(new Sign(R.drawable.img_f, signList.size(), "mean: 'F'"));
    signList.add(new Sign(R.drawable.img_g, signList.size(), "mean: 'G'"));
    signList.add(new Sign(R.drawable.img_h, signList.size(), "mean: 'H'"));
    signList.add(new Sign(R.drawable.img_i, signList.size(), "mean: 'I'"));
    signList.add(new Sign(R.drawable.img_j, signList.size(), "mean: 'J'"));
    signList.add(new Sign(R.drawable.img_k, signList.size(), "mean: 'K'"));
    signList.add(new Sign(R.drawable.img_l, signList.size(), "mean: 'L'"));
    signList.add(new Sign(R.drawable.img_m, signList.size(), "mean: 'M'"));
    signList.add(new Sign(R.drawable.img_n, signList.size(), "mean: 'N'"));
    signList.add(new Sign(R.drawable.img_o, signList.size(), "mean: 'O'"));
    signList.add(new Sign(R.drawable.img_p, signList.size(), "mean: 'P'"));
    signList.add(new Sign(R.drawable.img_q, signList.size(), "mean: 'Q'"));
    signList.add(new Sign(R.drawable.img_r, signList.size(), "mean: 'R'"));
    signList.add(new Sign(R.drawable.img_s, signList.size(), "mean: 'S'"));
    signList.add(new Sign(R.drawable.img_t, signList.size(), "mean: 'T'"));
    signList.add(new Sign(R.drawable.img_u, signList.size(), "mean: 'U'"));
    signList.add(new Sign(R.drawable.img_v, signList.size(), "mean: 'V'"));
    signList.add(new Sign(R.drawable.img_w, signList.size(), "mean: 'W'"));
    signList.add(new Sign(R.drawable.img_x, signList.size(), "mean: 'X'"));
    signList.add(new Sign(R.drawable.img_y, signList.size(), "mean: 'Y'"));
    signList.add(new Sign(R.drawable.img_z, signList.size(), "mean: 'Z'"));
    signList.add(new Sign(R.drawable.img_blank, 0, "mean: ''"));
}

@Override
public void onClick(String mean) {
    String result = textViewResult.getText() + mean;
    textViewResult.setText(result);
}
```

TextSignActivity

[illegible][illegible]

```

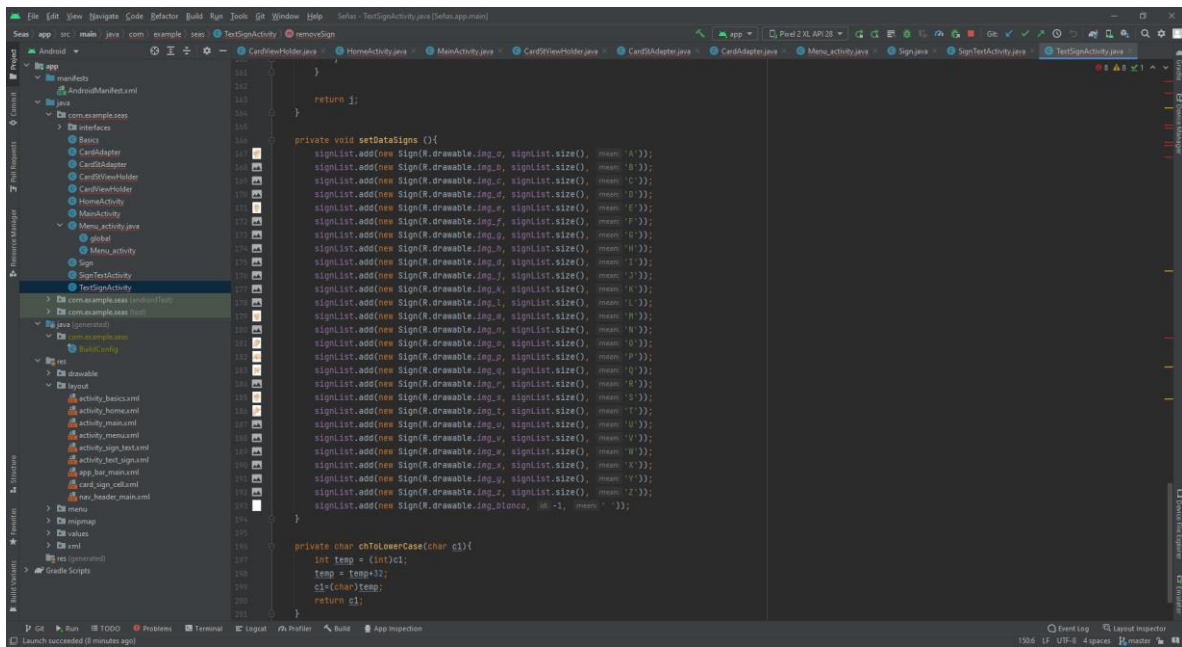
11  if (textFromMicro && stringLengthAfter - selectionEnd[0] > 1) { //char between text
12      lastChar = selectionEnd[0];
13  }
14  else { //Last char
15      lastChar = stringLengthAfter - 1;
16  }
17
18  for (int i = selectionStart[0]; i <= lastChar; i++) {
19      int id = getMeanId(getChar(str, i));
20      addSign(id, signTranslate, i);
21  }
22
23  if (stringLengthAfter < stringLengthBefore[0]) {
24      removeSign(selectionStart[0], selectionEnd[0], signTranslate);
25  }
26
27  //Update cards in recyclerview
28  binding.tvRecyclerView.setAdapter(new CardAdapter(signTranslate));
29  }
30
31  //Get microphone String
32  btnMicro.setOnClickListener(view -> {
33      signTranslate.clear();
34      Intent speechIntent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
35      speechIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
36      speechIntent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Say a text");
37      startActivityForResult(speechIntent, RECOGNIZER_RESULT);
38  });
39
40  @Override
41  protected void onActivityResult (int requestCode, int resultCode, @Nullable Intent data) {
42      if (requestCode == RECOGNIZER_RESULT && resultCode == RESULT_OK) {
43          ArrayList<String> matches = null;
44          if (data != null) {
45              matches = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
46          }
47          editText.setText(matches.get(0));
48      }
49  }

```

```

110
111  @Override
112  protected void onActivityResult (int requestCode, int resultCode, @Nullable Intent data) {
113      if (requestCode == RECOGNIZER_RESULT && resultCode == RESULT_OK) {
114          ArrayList<String> matches = null;
115          if (data != null) {
116              matches = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
117          }
118          editText.setText(matches.get(0));
119      }
120      super.onActivityResult(requestCode, resultCode, data);
121  }
122
123  private char getChar (String str, int nChar) {return str.charAt(nChar);}
124
125  private void addSign (int id, List<Sign> signTranslate, int tvPosition) {
126      signTranslate.add(tvPosition, signList.get(id));
127  }
128
129  private void removeSign (int selectionStart, int selectionEnd, List<Sign> signTranslate) {
130      if (selectionEnd > selectionStart) {
131          signTranslate.subList(selectionStart, selectionEnd).clear();
132      }
133      else {
134          signTranslate.remove(selectionEnd-1);
135      }
136  }
137
138  private int getMeanId (char c) {
139      int j=0;
140      for (int i = 0; i<signList.size(); i++){
141          char c1 = signList.get(i).mean;
142          if (c1 == c || Character.toLowerCase(c1) == c){
143              j++;
144              break;
145          }
146      }
147      return j;
148  }

```



Explicación de la codificación

Para realizar la explicación de forma más rápida y fluida se explicará a rasgos generales de la siguiente forma:

Pagina principal

Texto a seña

Seña a texto

Menú de señas básicas

Señas básicas

Pagina principal

Para empezar con la codificación de la pagina principal tuvimos que establecer antes cuantas secciones tendría nuestra aplicación de este modo se programaría tanto el menú respectivo a cada actividad, como el menú lateral, con el cual podríamos movernos entre la aplicación sin necesidad de regresar a la página principal para ello hicimos uso de un navigation item selected, el cual nos permite tener este menú desplegable, en este usamos un switch el cual nos permitirá saber cual de las secciones a sido clickeada y poder movernos hacia dicha actividad, en el cual tenemos los drawer dentro de un conten view el cual nos permite encapsular dichos menús y saber cuando son utilizados, en cuanto a la pagina principal este se trata de varios binding, que de igual forma consisten en el encapsulamiento de las actividades a través de un inten y mostrando sus imágenes respectivas

Texto a seña

Para la traducción de seña a texto utilizamos diferentes métodos para que este funcionara de forma correcta, empezamos por declara aspectos esenciales de nuestro código como lo fueron unos cuantos arrays, los cuales nos permitieron saber diferentes aspectos en cuanto a la evaluación del texto a ingresar, con ello

mandaremos a llamar a un text watcher con el cual podremos saber si el usuario esta escribiendo, no escribe o borra datos de nuestro texto, después con otro método evaluamos cuanto a cambiado nuestro texto y como este lo a hecho, si es que se eliminó, agrego o modifíco, ya sea en partes del inicio, en medio o final de este, tras esto, haremos aun mas uso de métodos los cuales nos permitirán evaluar con respecto a nuestro texto que seña es la que vamos a mostrar, un método estará para obtener el numero de caracteres de nuestro array, otro nos permitirá con estos caracteres evaluar a que letra hace referencia y con ello agregarlo y del mismo modo otro evaluara los caracteres y si estos no coinciden será eliminado, por último el método más sencillo es el de datos, el cual nos permitirá agregar a una lista, las diferentes señas a las cuales hace referencia cada una de las letras del abecedario

Seña a texto

Ya un poco mas sencillo contamos con la traducción que se hará de señas a texto, empezamos por crear los binding los cuales nos permitirán mostrar las imágenes de las señas y que el usuario pueda interactuar con ellas, de este modo y gracias a otro método de datos es que podemos agregar y presentar todas y cada una de las señas, tras ello y para poder evaluar que seña selecciono el usuario hacemos uso de un método adicional el cual consiste en evaluar a través de una comparación la carta seleccionada y asignarle este valor a un string, el cual después al txt con el que contamos se le agregara dicho string

Menú de señas básicas

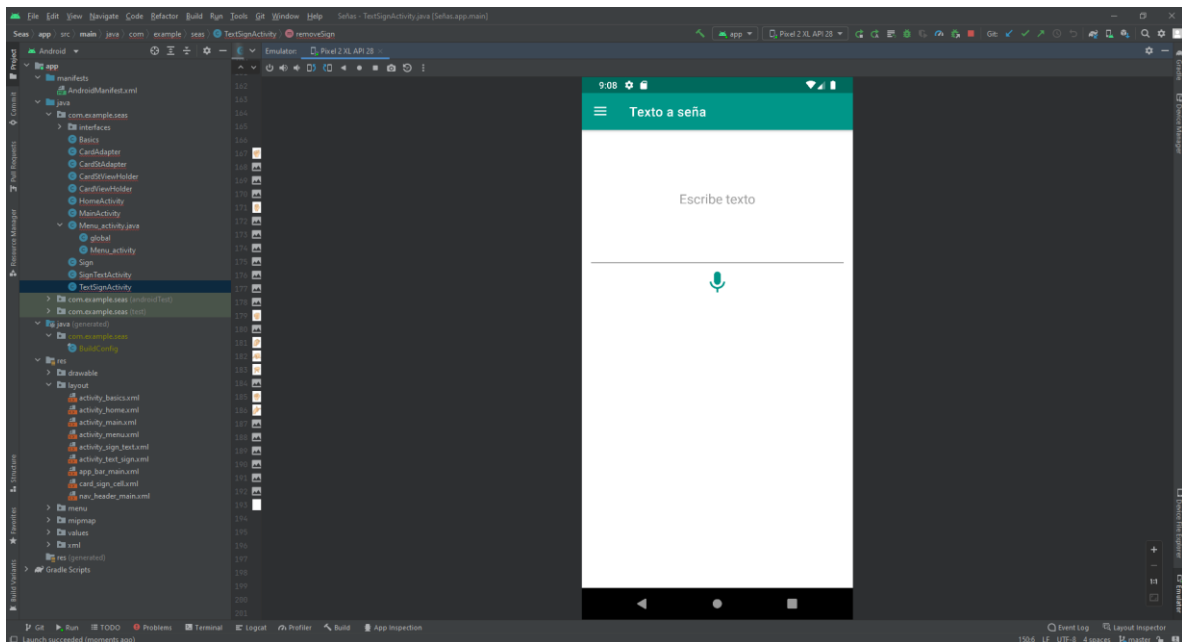
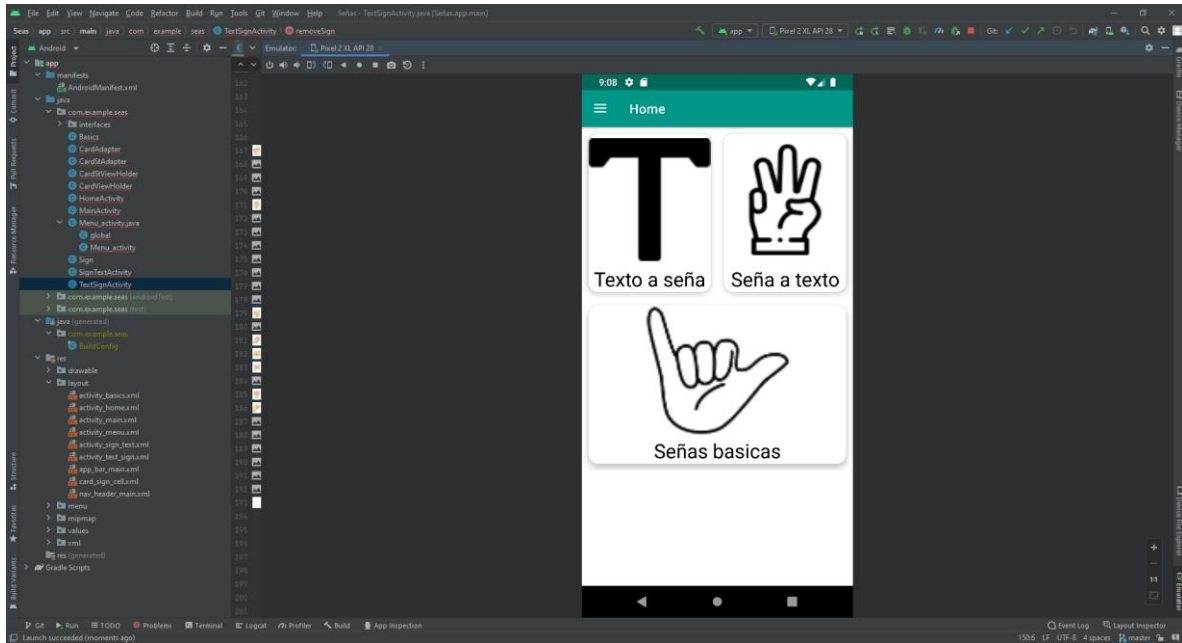
En esta sección fue sencilla su implementación ya que su simplicidad juega a su favor, consiste solamente en 3 botones los cuales es necesario declarar, además de esto, se declara un método global, esto para poder evaluar una variable que cambie a la hora de seleccionar un botón, con ello la información interna de todos y cada uno de los botones consistirá en un intent esto para poder movernos hacia la misma actividad ya que la diferencia será dada gracias a la variable, con ello en mente, lo único que cambiara es que el botón abecedario fijara dicha variable en 1,

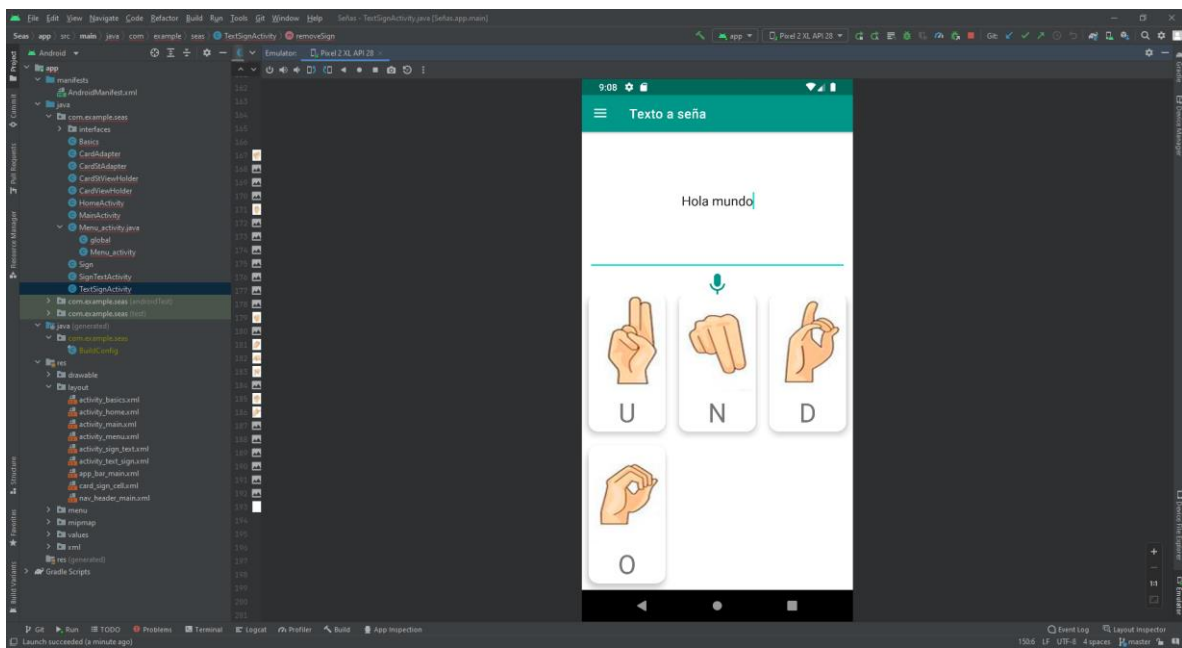
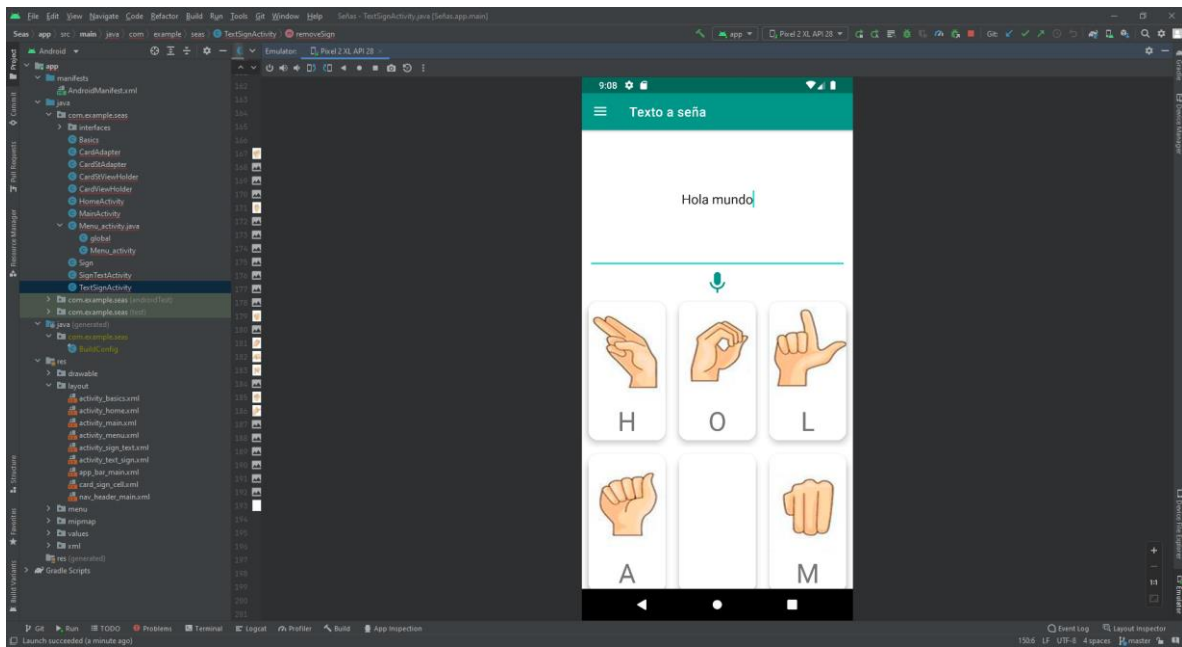
el botón meses y tiempo fijara la variable en 2 y el botón saludos fijara la variable en 3, con ello si nos encontráramos en la necesidad de agregar otra sección, solamente necesitaríamos declarar los botones, agregar el intent para cambiar de actividad y asignarle un valor a la variable (4 en adelante)

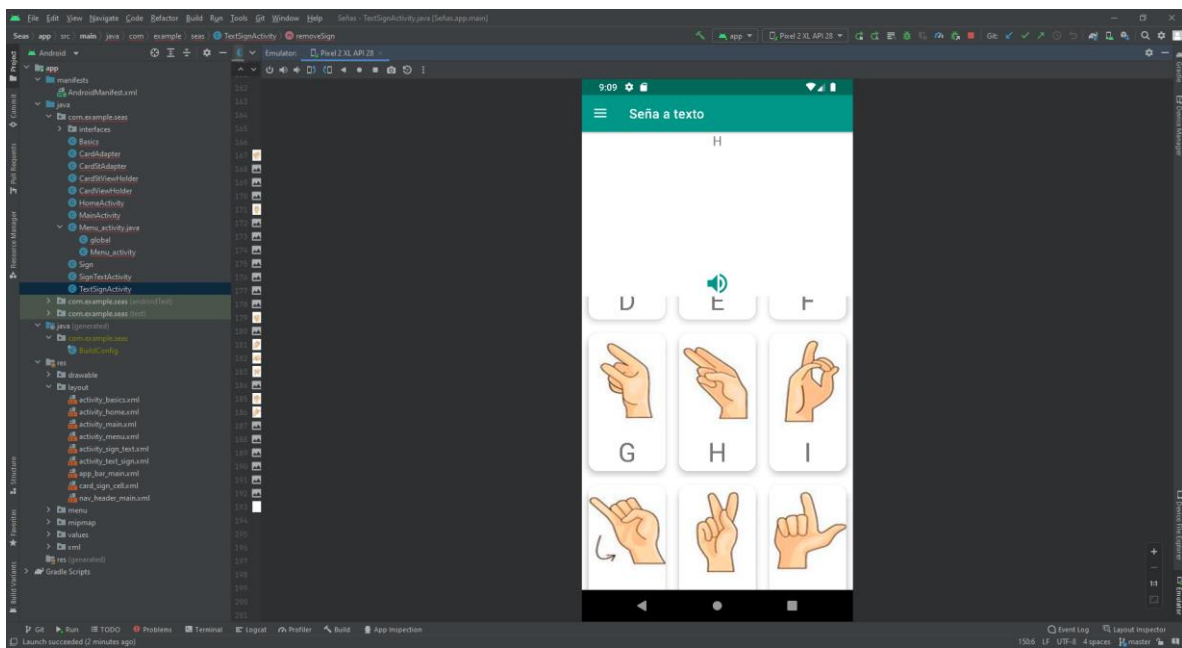
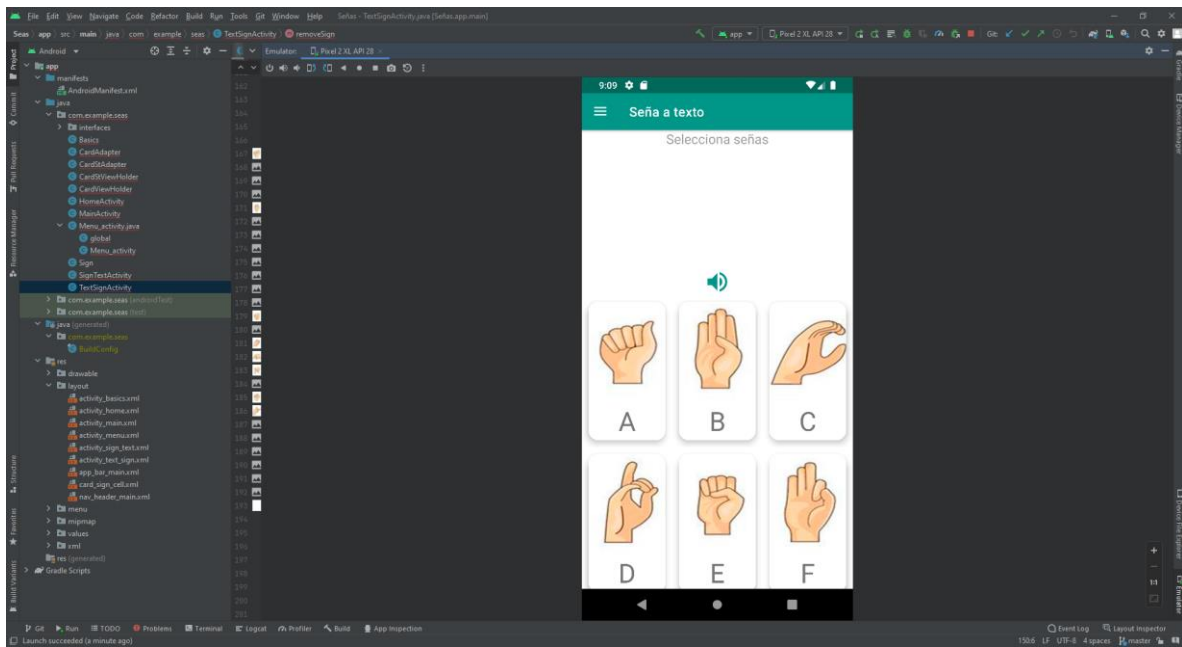
Señas básicas

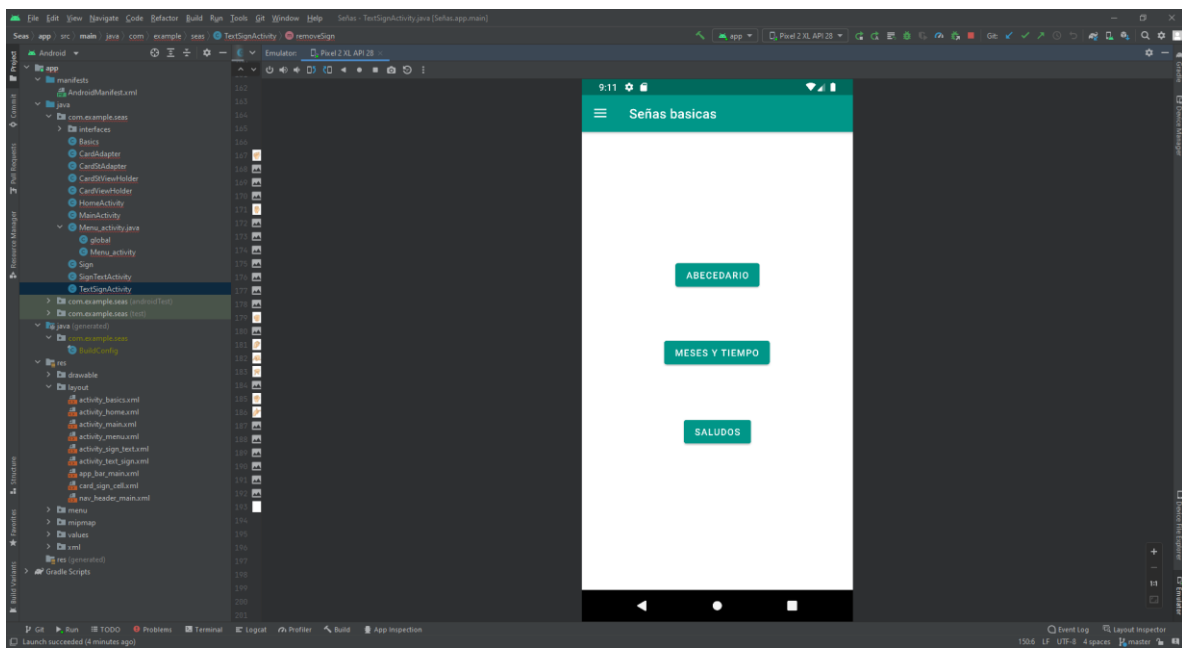
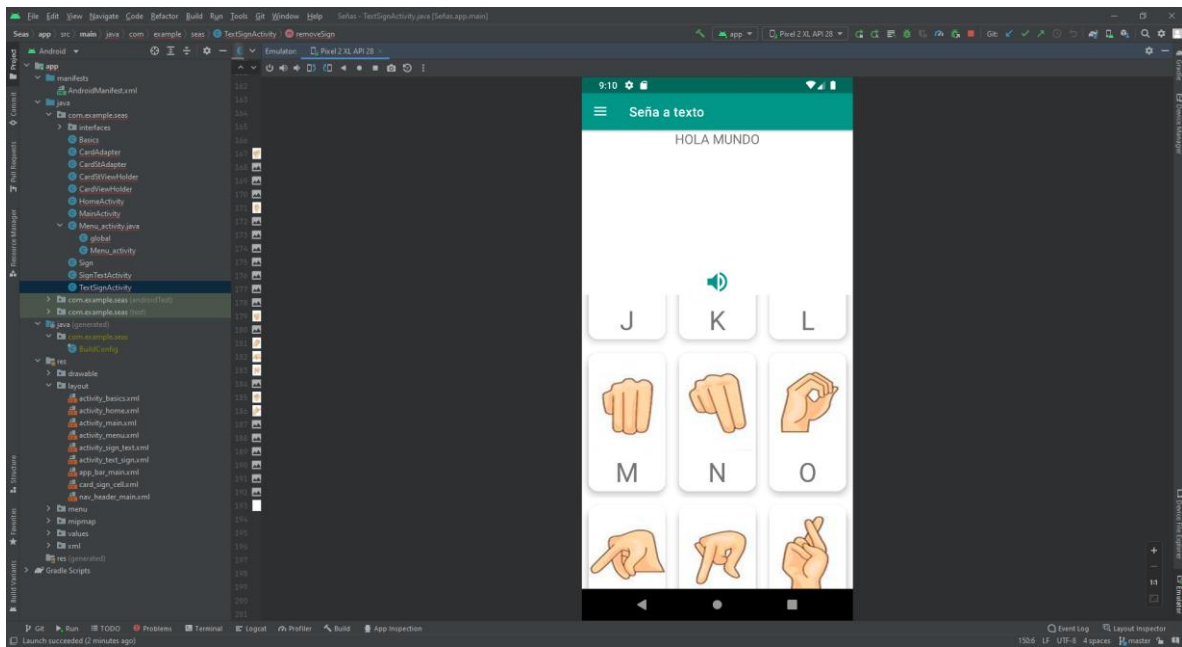
Para esta sección se programo de forma sencilla, se crearon 3 métodos casi iguales uno para el abecedario, uno para los meses y tiempo y otro para los saludos todos y cada uno contaba con características similares, la única diferencia era su contenido donde cada uno contendría sus respectivas señas gracias al método de llamado, para saber cual de estos métodos se utilizaría, usamos un switch en el cual hicimos uso de una variable global antes creada para evaluar que botón selecciono el usuario en el menú de señas, con ello si la variable contaba con un valor de 1 se presentarían las señas del abecedario, un valor de 2 se presentarían las señas de los meses y el tiempo y por ultimo un valor de 3 se presentarían los saludos, si quisiéramos agregar mas señas a alguna de las secciones, bastaría con colocar solamente una línea y siendo ligeramente modificada para mandar a llamar a la imagen respectiva de esta para que podamos agregarla, aunque si quisiéramos agregar una sección adicional, si bien el trabajo de modificación es un poco mayor, en realidad tampoco es algo muy complicado, seria cuestión de crear el método respectivo, agregar las señas que queremos encontrar en dicho método, agregarlo al switch y agregarlo al menú de selección

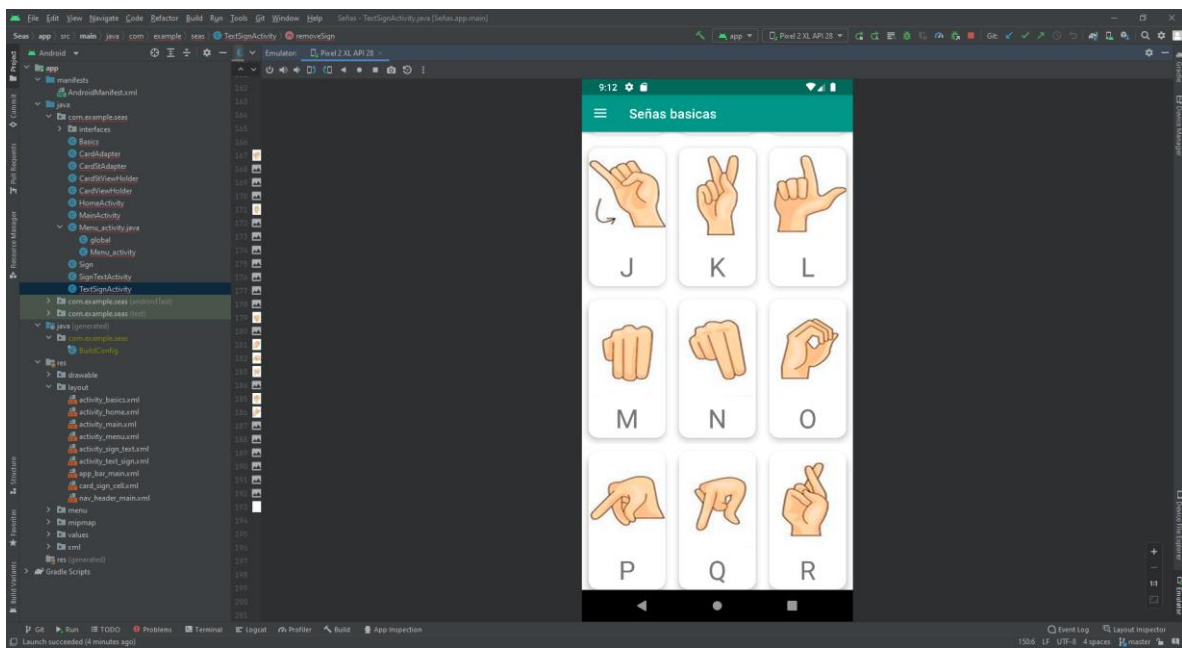
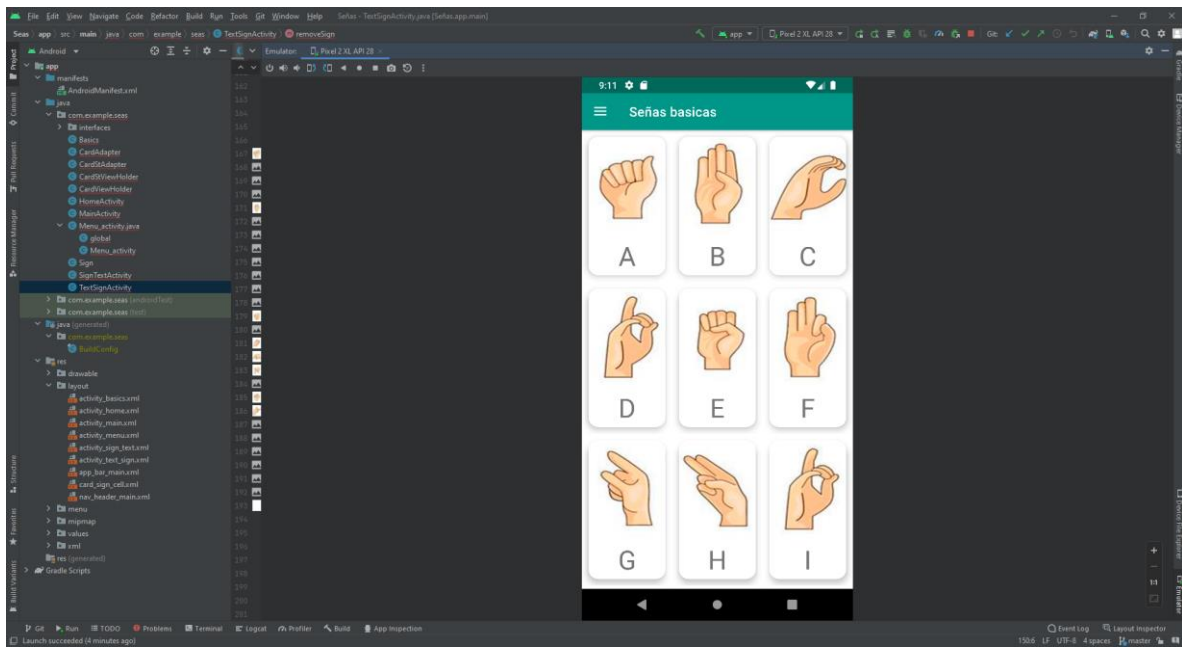
Imágenes de la ejecución

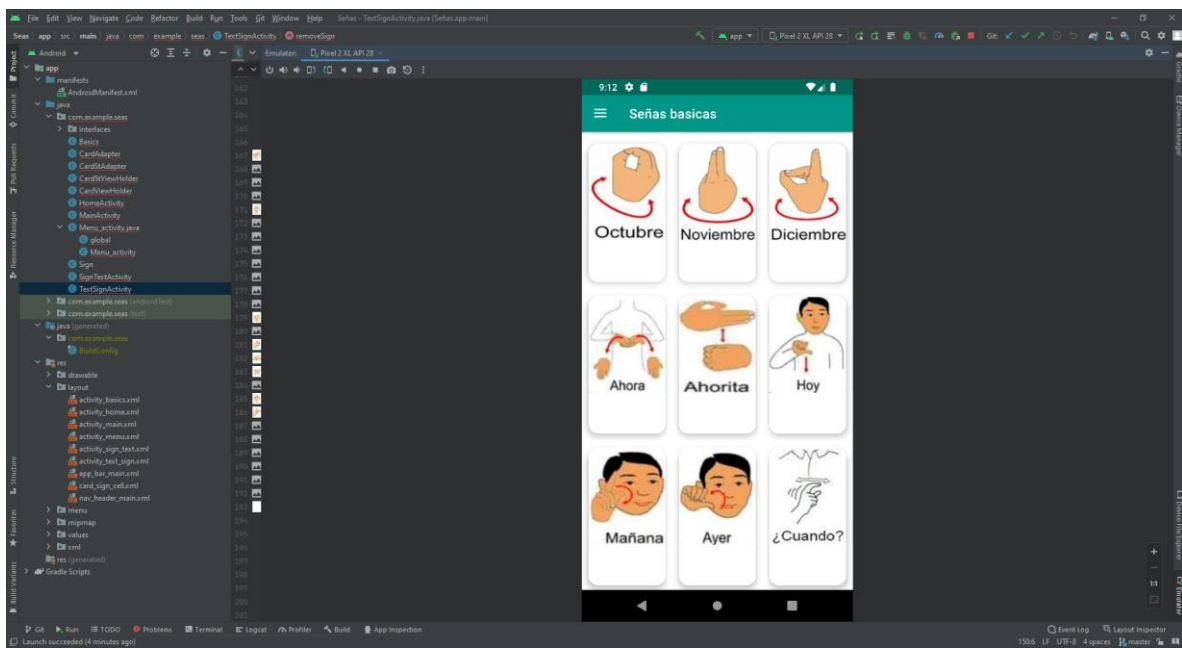
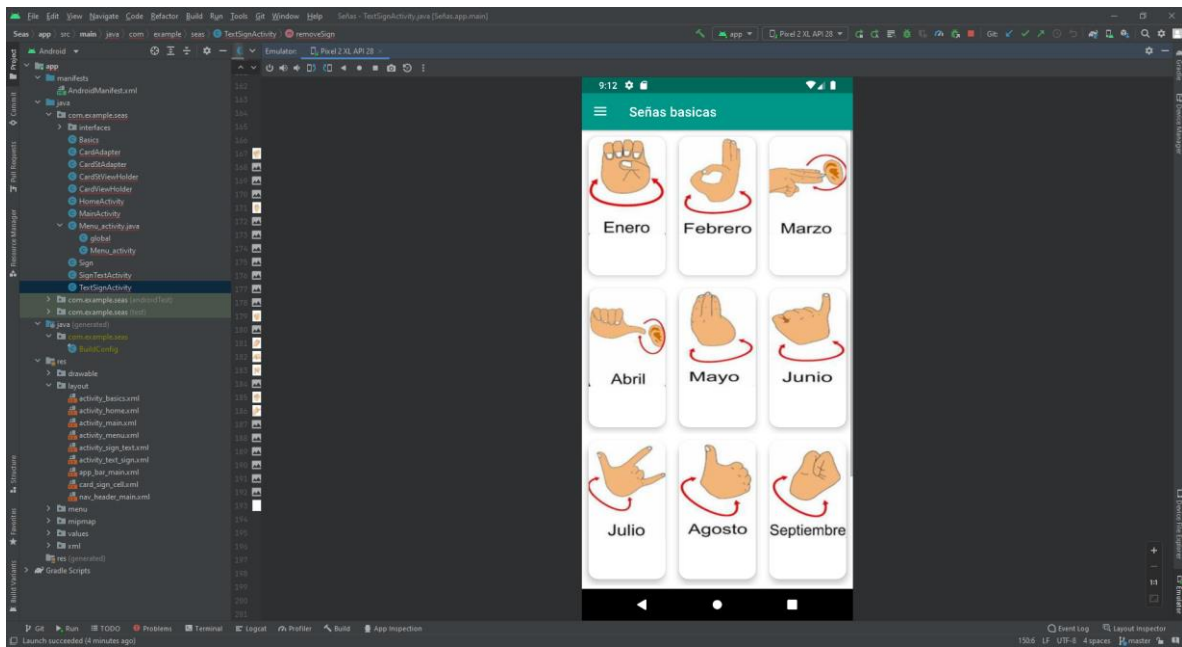


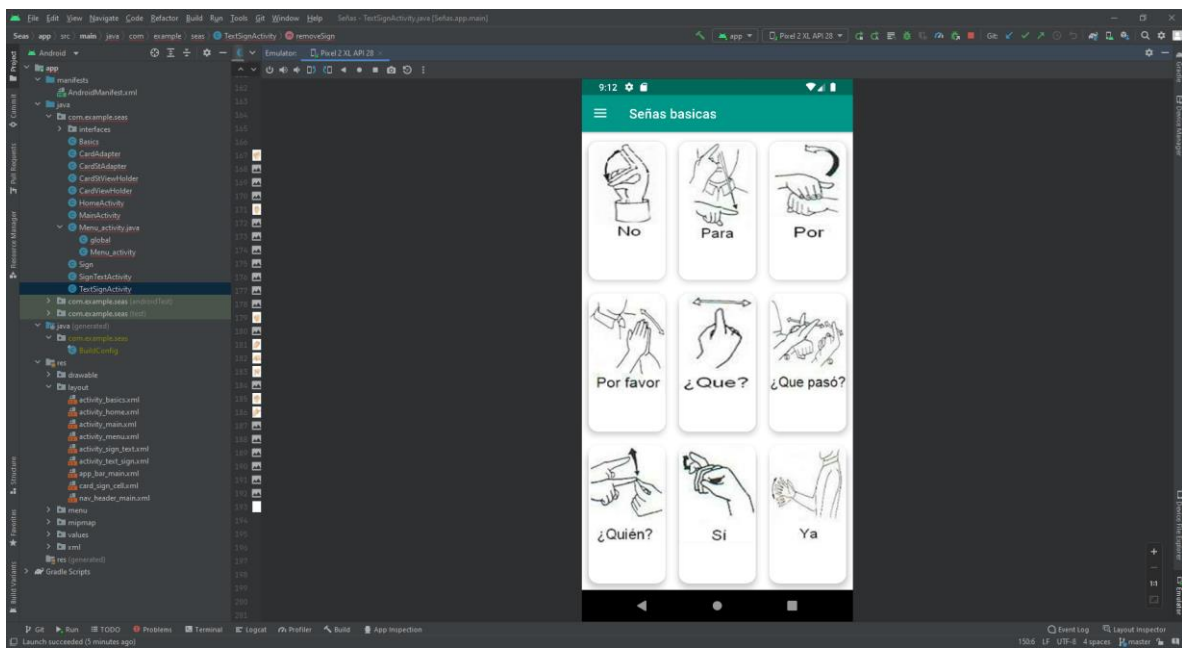
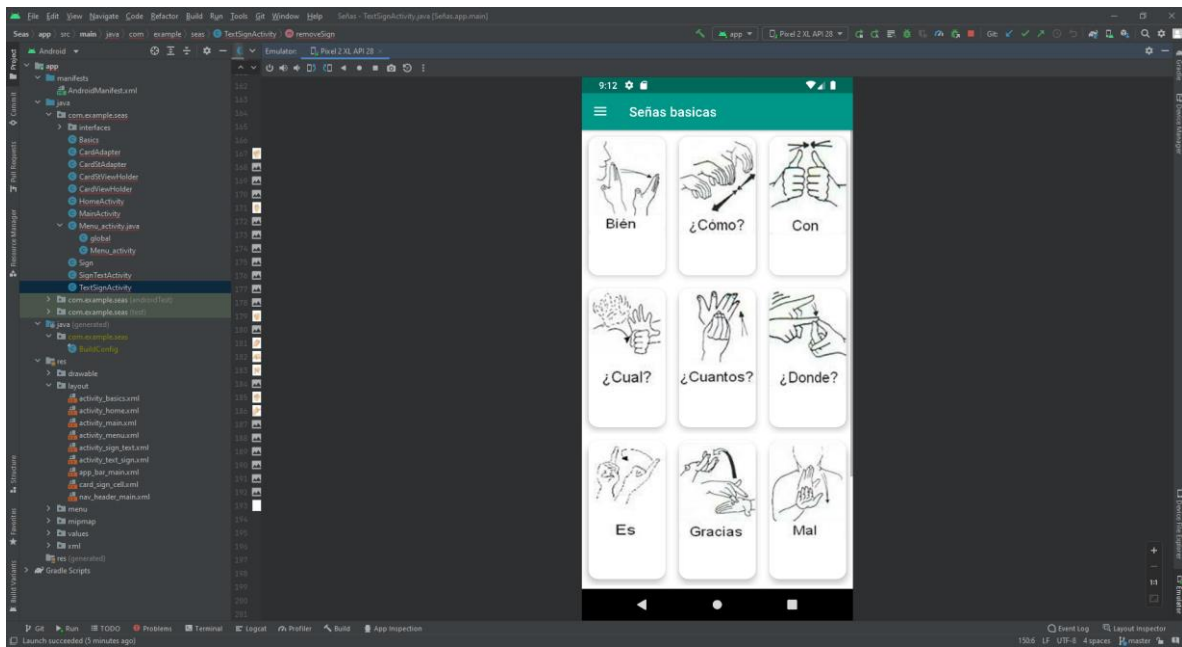


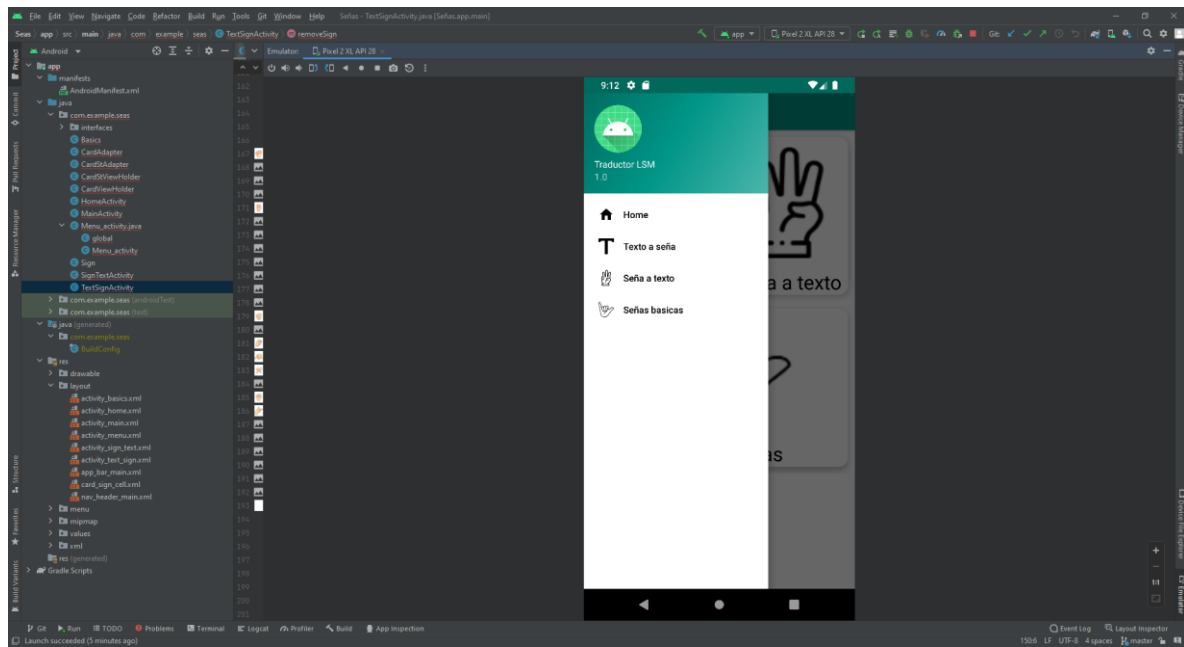












Resultados

Dentro de nuestra practica consideramos tener resultados bastante favorables a lo que en la versión 1.0 corresponde, ya que al final de esta, son casi nulos los fallos que se tienen que arreglar y son mas cuestiones del aspecto visual y de navegación los cuales se podrían mejorar, siendo estos los próximos a modificar es versiones posteriores, de este modo también consideramos que como se menciona al principio la idea principal es poder realizar una traducción muy básica, esto con el fin de que si al menos la comunicación no sería la más específica/adecuada, esta sería al menos la necesaria para poder llevar a cabo un intercambio de ideas muy básicos, lo cual es muchísimo mejor a que si no nos pudiéramos comunicar, siendo esta la principal razón de la realización de esta actividad, del mismo modo al ser interactiva y sencilla de usar consideramos que tanto adultos como niños pueden hacer uso de esta aplicación, para ayudarse al día a día o incluso para sostener una conversación sencilla.