TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE CHETUMAL

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Programación de Dispositivos Móviles

Calve de la Asignatura: SSC-1019

SATCA: 2-2-4

Evaluación de Final de curso

NOMBRE DEL PROFESOR: Isaías May Canche

ALUMNO: Aranda Cen Jeancarlo Antonio

No. CONTROL: 17390318

GRUPO: I8U

12 DE JULIO DEL 2021

MARZO-JULIO 2021

Índice

Instrucciones

Realizar programa en Android Estudio para Calcular la CURP

Introducción

En este proyecto se realizará una aplicación móvil en Android Studio con el fin de obtener datos del usuario para armar la curp la cual es unica para cada individuo el país y por si llegara a ser el caso dos últimos dígitos especiales homoclave los cuales pueden alternarse para evitar la duplicación de curp

Detalles

Cómo se hace el Curp:

Inicial y primera vocal del apellido paterno (Ejemplo: **Fe**rnandez=**Fe**)

Inicial del apellido materno (Perez=**P**)

Inicial del nombre principal (Manuel=**M**)

Fecha de nacimiento (en el siguiente formato “1989-02-04” en 6 dígitos = **890204**)

Sexo (**H** si es hombre, M para mujer)

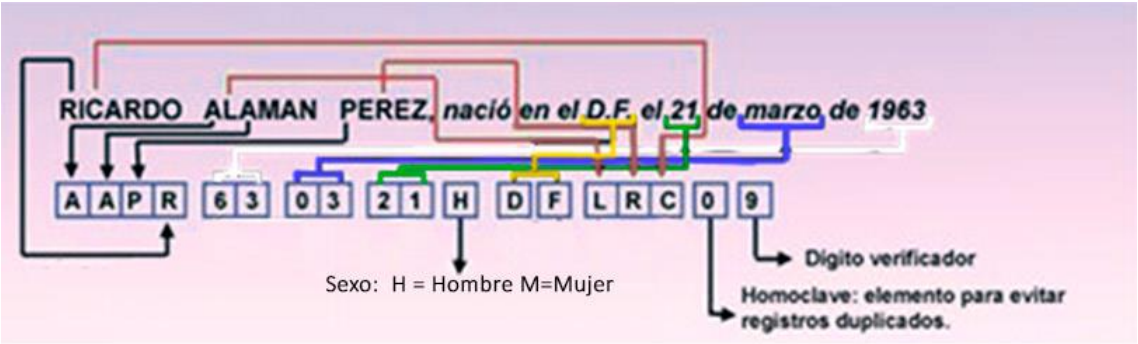
Abreviatura del lugar de nacimiento (Ver lista de abreviaturas de cada estado de México aquí) (Ejemplo: Aguascalientes=**AS**)

Primera consonante no inicial del apellido del padre (Fe**r**nandez=**R**)

Primera consonante no inicial del apellido de la madre (Pe**r**ez=**R**)

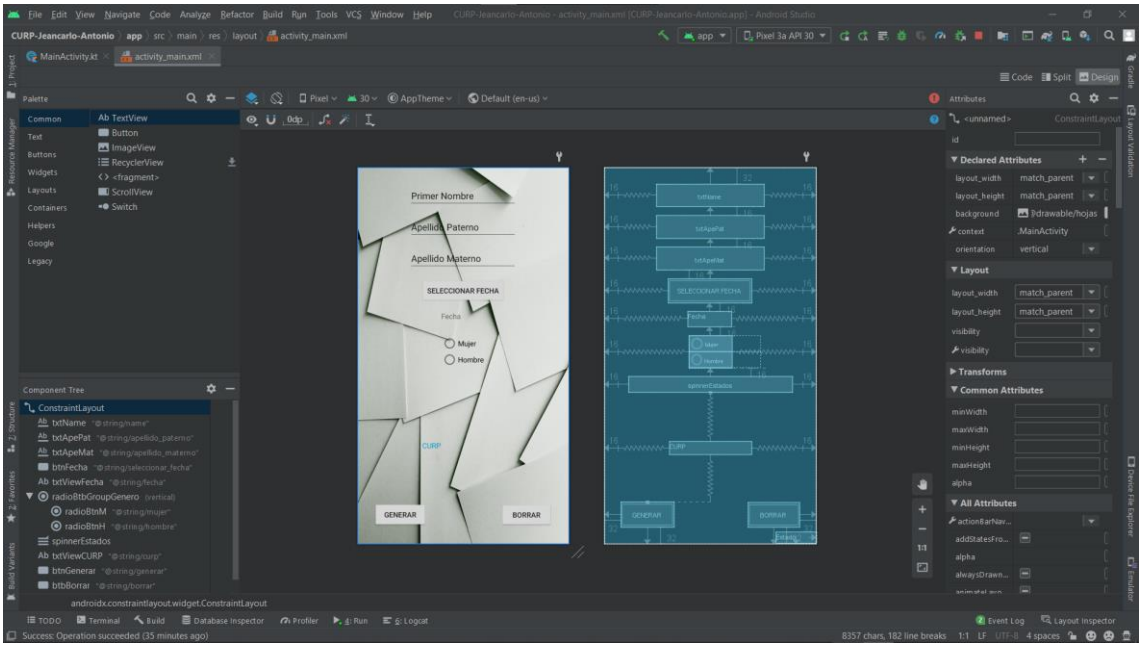
Primera consonante no inicial del nombre del consultor de la curp (Ma**n**uel=**N**)

Dos dígitos llamados Homoclave (Genera tu homoclave propia).

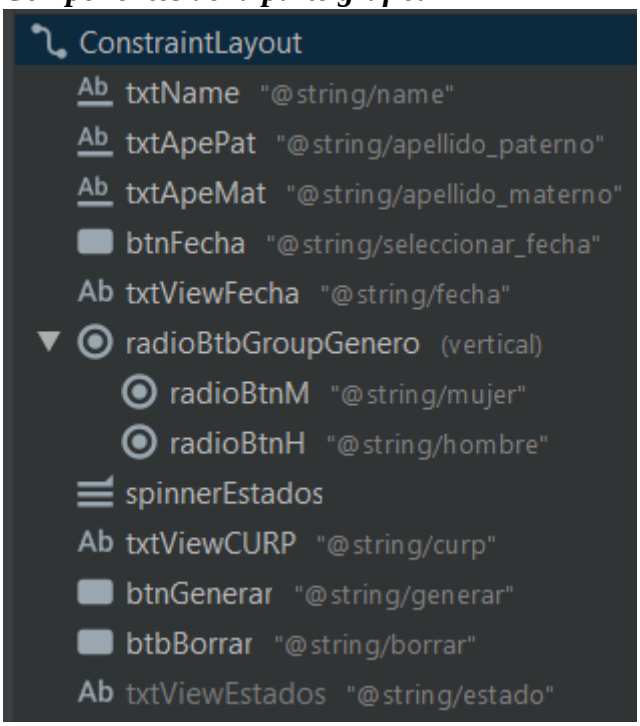


CURP

Vista del usuario



Componentes de la parte grafica



La parte gráfica de la aplicación contiene diferentes componentes tales como de texto donde ingresaremos datos del usuario, botones 1 en particular para seleccionar la fecha en el cual desplegaremos un detepicker para elegir a de manera de manera más eficiente un grupo de radio buttons con la finalidad de elegir entre el género, spinner en el cual mostrará todos los Estados de México y por último un campo de texto que sólo es demostrativo en el cual nos servirá para mostrar la curp resultante y sus debidos botones de general y borrar

Código

```
package com.example.curp_jeancarlo_antonio

import android.app.DatePickerDialog
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.*
import kotlinx.android.synthetic.main.activity_main.*
import java.util.*

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)


//*************************************************************************
*************->
        val calendario = Calendar.getInstance()
        val DD = calendario.get(Calendar.DAY_OF_MONTH)
        val MM = calendario.get(Calendar.MONTH)
        val AAAA = calendario.get(Calendar.YEAR)

        btnFecha.setOnClickListener {
            var datePickD =
DatePickerDialog(this,DatePickerDialog.OnDateSetListener { view, aaaa, mm, dd
->
                //se le suma un 1 al mes por que empiezan desde 0
                var mes = (mm+1).toString()
                //agrega un cero al valor del mes si este es menor a 10
                if (mes.toInt()<10){
                    mes = "0" + mes
                }
                //agrega un cero al valor del dia si este es menor a 10
                var dia = (dd).toString()
                if (dia.toInt()<10){
                    dia = "0" + dia
                }
                txtViewFecha.setText(""+ dia + "/" + (mes) + "/" +
aaaa)},AAAA, MM, DD)
            datePickD.show()
        }

        var genero = ""

        radioBtbGroupGenero.setOnCheckedChangeListener { group, i ->
            if (i == R.id.radioBtnH)
                genero = radioBtnH.text[0].toString()
            Toast.makeText(this,genero,Toast.LENGTH_SHORT).show()

            if (i == R.id.radioBtnM)
                genero = radioBtnM.text[0].toString()
            Toast.makeText(this,genero,Toast.LENGTH_SHORT).show()
        }

        var arrayEstados = arrayOf("AS-Aguascalientes","BC-Baja
California","BS-Baja California Sur","CC-Campeche","CL-Coahuila","CM-Colima",
```

```
"CS-Chiapas","CH-Chihuahua","DF-Distrito Federal","DG-Durango","GT-
Guanajuato","GR-Guerrero","HG-Hidalgo","JC-Jalisco","MC-México","MN-
Michoacán","MS-Morelos","NT-Nayarit", "NL-Nuevo León","OC-Oaxaca","PL-
Puebla","QT-Querétaro","QR-Quintana Roo","SP-San Luis Potosí","SL-
Sinaloa","SR-Sonora","TC-Tabasco","TS-Tamaulipas", "TL-Tlaxcala","VZ-
Veracruz","YN-Yucatan","ZS-Zacatecas","NE-Nacido en el Extrenjero")

        val arrayAdapter =
ArrayAdapter(this@MainActivity,android.R.layout.simple_spinner_dropdown_item,
arrayEstados)
        spinnerEstados.adapter = arrayAdapter

        spinnerEstados.onItemSelectedListener = object
:AdapterView.OnItemSelectedListener {
            override fun onItemSelected(
                parent: AdapterView<*>?,
                view: View?,
                position: Int,
                id: Long
            ) {
                val estado = (arrayEstados[position])
                txtViewEstados.setText(estado)
            }

            override fun onNothingSelected(parent: AdapterView<*>?) {
            }
        }

        val mutableValues =
mutableListOf("0","1","2","3","4","5","6","7","8","9")

        btnGenerar.setOnClickListener {
            val name = txtName.text;
            val consonanteNombre = SinVocal(name.toString())
            var segundaConsonateNombre16 = ""

            if (name[0].toLowerCase() == 'a'|| name[0].toLowerCase() == 'e'||
name[0].toLowerCase() == 'i'|| name[0].toLowerCase() == 'o'||
name[0].toLowerCase() == 'u' ){
                segundaConsonateNombre16 =
consonanteNombre[0].toString().toUpperCase()
            }
            else{
                segundaConsonateNombre16 =
consonanteNombre[1].toString().toUpperCase()
            }

            val apellidoPaterno = txtApePat.text
            var apellidoVocal = SinConsonante(apellidoPaterno.toString())
            var primerVocalPat2 = ""

            if (apellidoPaterno[0].toLowerCase() == 'a'||
apellidoPaterno[0].toLowerCase() == 'e'|| apellidoPaterno[0].toLowerCase() ==
'i'|| apellidoPaterno[0].toLowerCase() == 'o'||
apellidoPaterno[0].toLowerCase() == 'u' ){
                primerVocalPat2 = apellidoVocal[1].toString().toUpperCase()
            }
            else{
                primerVocalPat2 = apellidoVocal[0].toString().toUpperCase()
```

```
            }

            val consonantePaterno = SinVocal(apellidoPaterno.toString())
            var segundaConsonatePat14 = ""

            if (apellidoPaterno[0].toLowerCase() == 'a'||
apellidoPaterno[0].toLowerCase() == 'e'|| apellidoPaterno[0].toLowerCase() ==
'i'|| apellidoPaterno[0].toLowerCase() == 'o'||
apellidoPaterno[0].toLowerCase() == 'u' ){
                segundaConsonatePat14 =
consonantePaterno[0].toString().toUpperCase()
            }
            else{
                segundaConsonatePat14 =
consonantePaterno[1].toString().toUpperCase()
            }

            val apellidoMaterno = txtApeMat.text
            val consonanteMaterno = SinVocal(apellidoMaterno.toString())
            var segundaConsonateMat15 = ""
            if (apellidoMaterno[0].toLowerCase() == 'a'||
apellidoMaterno[0].toLowerCase() == 'e'|| apellidoMaterno[0].toLowerCase() ==
'i'|| apellidoMaterno[0].toLowerCase() == 'o'||
apellidoMaterno[0].toLowerCase() == 'u' ){
                segundaConsonateMat15 =
consonanteMaterno[0].toString().toUpperCase()
            }
            else{
                segundaConsonateMat15 =
consonanteMaterno[1].toString().toUpperCase()
            }

            val fecha = txtViewFecha.text.toString()

            if(name.length == 0 || apellidoPaterno.length == 0 ||
apellidoMaterno.length == 0){
                Toast.makeText(this,"No ah ingresado todos sus
datos",Toast.LENGTH_SHORT).show()
            }
            else{
                var primerPat1 = apellidoPaterno[0].toString().toUpperCase()
                var primerMat3 = apellidoMaterno[0].toString().toUpperCase()
                var primerNombre4 = name[0].toString().toUpperCase()
                var decada5 = fecha[8]
                var año6 = fecha[9]
                var mes7 = fecha[3]
                var mes8 = fecha[4]
                var dia9 = fecha[0]
                var dia10 = fecha[1]
                var genero11 = genero
                var estado12 = txtViewEstados.text[0]
                var estado13 = txtViewEstados.text[1]
                val valor17 = mutableValues.random()
                val valor18 = mutableValues.random()

                txtViewCURP.text = primerPat1 + primerVocalPat2 + primerMat3
+ primerNombre4 + decada5 + año6 + mes7 + mes8 + dia9 + dia10 + genero11 +
estado12 + estado13 + segundaConsonatePat14 + segundaConsonateMat15 +
segundaConsonateNombre16 + valor17 + valor18
```

```
            }
        }

        btbBorrar.setOnClickListener {
            var nombre = findViewById<EditText>(R.id.txtName)
            nombre.text.clear()
            var apePat = findViewById<EditText>(R.id.txtApePat)
            apePat.text.clear()
            var apeMat = findViewById<EditText>(R.id.txtApeMat)
            apeMat.text.clear()
            var fecha = findViewById<TextView>(R.id.txtViewFecha)
            fecha.setText("Fecha")
            var estado = findViewById<TextView>(R.id.txtViewEstados)
            estado.setText("Estado")
            var curp = findViewById<TextView>(R.id.txtViewCURP)
            curp.setText("CURP")
        }

//*************************************************************************
************<-
    }


//*************************************************************************
**
    fun SinConsonante( text: String ): String {
        val resultado = StringBuilder()
        for ( char in text ) {
            if ( !"bcdfghjklmnñpqrstvwxyz".contains(char.toLowerCase()) ) {

                resultado.append( char )
            }
        }
        return resultado.toString()
    }

    fun SinVocal( text: String ): String {
        val resultado = StringBuilder()
        for ( char in text ) {
            if ( !"aeiou".contains(char.toLowerCase()) ) {

                resultado.append( char )
            }
        }
        return resultado.toString()
    }

//*************************************************************************
*****************<-
}
```

Explicación de funcionamiento de las líneas de código

Creamos 4 variables una en particular que obtiene la instancia del calendario y otras 3 para sacar el día mes y el año de dicho calendario también tenemos un botón que al momento de ser presionado manda a crear un nuevo detePicker que esto internamente lo hace Android Studio pero una vez tengamos este resultado lo primero que se hace creamos una variable mes la cual le vamos a sumar 1 ya que empiezan desde cero y la convertimos a tipo texto y por consiguiente como la curp nos pide dos dígitos para el mes o para el día validamos si el valor de éste es menor a 10 y si es este el caso le pondremos un cero delante y así de esta manera igual al día y por último nuestra etiqueta de texto la rellenamos con el acomodo de la nueva fecha que hemos creado día mes y año y por último este la mostramos

```
val calendario = Calendar.getInstance()
val DD = calendario.get(Calendar.DAY_OF_MONTH)
val MM = calendario.get(Calendar.MONTH)
val AAAA = calendario.get(Calendar.YEAR)

btnFecha.setOnClickListener {  it: View!
    var datePickD = DatePickerDialog( context: this,DatePickerDialog.OnDateSetListener { view, aaaa, mm, dd ->
        //se le suma un 1 al mes por que empiezan desde 0
        var mes = (mm+1).toString()
        //agrega un cero al valor del mes si este es menor a 10
        if (mes.toInt()<10){
            mes = "0" + mes
        }
        //agrega un cero al valor del dia si este es menor a 10
        var dia = (dd).toString()
        if (dia.toInt()<10){
            dia = "0" + dia
        }
        txtViewFecha.setText(""+ dia + "/" + (mes) + "/" + aaaa)},AAAA, MM, DD)
    datePickD.show()
}
```

Queremos una variable género vacía en este caso de tipo texto la cual posteriormente utilizaremos y ahora tenemos un evento que escucho cuando sea dado el alguno de los dos radio botones en este caso son dos 1 para el género hombre y otro para el género mujer en el cual asignamos una id que distingue entre cada 1 y se le asigna un valor de texto con la letra h con la letra m

```
var genero = ""

radioBtbGroupGenero.setOnCheckedChangeListener { group, i ->
    if (i == R.id.radioBtnH)
        genero = radioBtnH.text[0].toString()
    Toast.makeText( context: this,genero,Toast.LENGTH_SHORT).show()

    if (i == R.id.radioBtnM)
        genero = radioBtnM.text[0].toString()
    Toast.makeText( context: this,genero,Toast.LENGTH_SHORT).show()
}
```

Creamos el arreglo de Estados el cual es un arreglo de texto el cual contiene todas las abreviaturas de los Estados de México el cual lo recorremos con un spinner para esto creamos la variable adaptador el cual le asignamos todo el arreglo de Estados y tenemos un evento al seleccionar un ítem el cual no tiene su ID la vista en la posición la cual le asignaremos en la parte gráfica de la aplicación la etiqueta de Estados como tu texto

```
var arrayEstados = arrayOf("AS-Aguascalientes","BC-Baja California","BS-Baja California Sur","CC-Campeche","CL-Coahuila")

val arrayAdapter = ArrayAdapter( context: this@MainActivity,android.R.layout.simple_spinner_dropdown_item,arrayEstados)
spinnerEstados.adapter = arrayAdapter

spinnerEstados.onItemSelectedListener = object :AdapterView.OnItemSelectedListener {
    override fun onItemSelected(
        parent: AdapterView<*>?,
        view: View?,
        position: Int,
        id: Long
    ) {
        val estado = (arrayEstados[position])
        txtViewEstados.setText(estado)
    }

    override fun onNothingSelected(parent: AdapterView<*>?) {
    }
}
```

Aquí igual creamos una variable de un arreglo en este caso de tipo texto la cual contiene todos los números naturales

```
val mutableValues = mutableListOf("0","1","2","3","4","5","6","7","8","9")
```

Para el botón borrar cuando sea presionado tenemos que va a limpiar todos los campos en este caso el campo de nombre el apellido paterno apellido materno y al de fecha estado y el curso les asignará estas respectivos nombres

```
btbBorrar.setOnClickListener {  it: View!
    var nombre = findViewById<EditText>(R.id.txtName)
    nombre.text.clear()
    var apePat = findViewById<EditText>(R.id.txtApePat)
    apePat.text.clear()
    var apeMat = findViewById<EditText>(R.id.txtApeMat)
    apeMat.text.clear()
    var fecha = findViewById<TextView>(R.id.txtViewFecha)
    fecha.setText("Fecha")
    var estado = findViewById<TextView>(R.id.txtViewEstados)
    estado.setText("Estado")
    var curp = findViewById<TextView>(R.id.txtViewCURP)
    curp.setText("CURP")
}
```

Por último tenemos una función en este caso le nombramos sin consonante de acuerdo obtendrá un una regla de tipo texto en este caso ya sea el nombre fue el apellido y con esto volvemos a construir una cadena en el cual tenemos un For que recorre mientras haya variables de tipo texto que tenemos que si encuentra alguna letra consonante las

eliminará y sólo nos quedarán las vocales y este última arreglo lo devolverá es de igual manera para la otra función sin vocal pero en este caso sólo devolverá consonantes

```
fun SinConsonante( text: String ): String {
    val resultado = StringBuilder()
    for ( char in text ) {
        if ( !"bcdfghjklmnñpqrstvwxyz".contains(char.toLowerCase()) ) {

            resultado.append( char )
        }
    }
    return resultado.toString()
}

fun SinVocal( text: String ): String {
    val resultado = StringBuilder()
    for ( char in text ) {
        if ( !"aeiou".contains(char.toLowerCase()) ) {

            resultado.append( char )
        }
    }
    return resultado.toString()
}
```

Comentarios acerca de la operación de la creación del curp

Aquí explicaremos todos los precios el botón generala el último como la ordena oh muestra

```
btnGenerar.setOnClickListener {  it: View!
    val name = txtName.text;
    val consonanteNombre = SinVocal(name.toString())
    var segundaConsonateNombre16 = ""

    if (name[0].toLowerCase() == 'a'|| name[0].toLowerCase() == 'e'|| name[0].toLowerCase() == 'i'|| name[0].toLowerCase() == 'o'|| name[0].toLowerCase() == 'u' ){
        segundaConsonateNombre16 = consonanteNombre[0].toString().toUpperCase()
    }
    else{
        segundaConsonateNombre16 = consonanteNombre[1].toString().toUpperCase()
    }

    val apellidoPaterno = txtApePat.text
    var apellidoVocal = SinConsonante(apellidoPaterno.toString())
    var primerVocalPat2 = ""

    if (apellidoPaterno[0].toLowerCase() == 'a'|| apellidoPaterno[0].toLowerCase() == 'e'|| apellidoPaterno[0].toLowerCase() == 'i'|| apellidoPaterno[0].toLowerCase() =
        primerVocalPat2 = apellidoVocal[1].toString().toUpperCase()
    }
    else{
        primerVocalPat2 = apellidoVocal[0].toString().toUpperCase()
    }

    val consonantePaterno = SinVocal(apellidoPaterno.toString())
    var segundaConsonatePat14 = ""

    if (apellidoPaterno[0].toLowerCase() == 'a'|| apellidoPaterno[0].toLowerCase() == 'e'|| apellidoPaterno[0].toLowerCase() == 'i'|| apellidoPaterno[0].toLowerCase() =
        segundaConsonatePat14 = consonantePaterno[0].toString().toUpperCase()
    }
    else{
        segundaConsonatePat14 = consonantePaterno[1].toString().toUpperCase()
    }
```

Creamos la variable nombre que es el atributo de texto que obtenemos de la parte gráfica la variable consonante nombre mandar a llamar a la función sin vocal la cual le regresará la cadena solamente con consonantes, la variable segunda consonante del nombre es la posición 16 esta se probará que no empiece con vocal para que agarre la primera posición de la regla de consonantes si no obtendrá la segunda posición

```
val name = txtName.text;
val consonanteNombre = SinVocal(name.toString())
var segundaConsonateNombre16 = ""

if (name[0].toLowerCase() == 'a'|| name[0].toLowerCase() == 'e'|| name[0].toLowerCase() == 'i'|| name[0].toLowerCase() == 'o'|| name[0].toLowerCase() == 'u' ){
    segundaConsonateNombre16 = consonanteNombre[0].toString().toUpperCase()
}
else{
    segundaConsonateNombre16 = consonanteNombre[1].toString().toUpperCase()
}
```

De igual manera como explicamos pedimos el atributo texto de la parte gráfica del apellido paterno y con esto creamos una variable que si me apellido paterno esta misma es enviada a la función sin consonante la cual nos volverá sólo las vocales y con eso crearemos la variable apellido vocal ahora se comprueba que la primera letra del apellido paterno sea vocal y es que entonces elegir a la segunda posición sí no elegir a la primera vocal que encuentre de la variable apellido vocal

```
val apellidoPaterno = txtApePat.text
var apellidoVocal = SinConsonante(apellidoPaterno.toString())
var primerVocalPat2 = ""

if (apellidoPaterno[0].toLowerCase() == 'a'|| apellidoPaterno[0].toLowerCase() == 'e'|| apellidoPaterno[0].toLowerCase() == 'i'|| apellidoPaterno[0].toLowerCase()
    primerVocalPat2 = apellidoVocal[1].toString().toUpperCase()
}
else{
    primerVocalPat2 = apellidoVocal[0].toString().toUpperCase()
}
```

La variable apellido paterno lo mandamos como texto a la función sin vocal la cual nos devolverá una cadena con las consonantes de esta es así como crearemos la variable consonante paterno y con esta probaremos que la variable apellido paterno empieza con alguna vocal para obtener de la variable consonante paterno la primera posición si no obtendrá la segunda y se le asignarán a la variable segunda consonante que será la posición núm14

```
val consonantePaterno = SinVocal(apellidoPaterno.toString())
var segundaConsonatePat14 = ""

if (apellidoPaterno[0].toLowerCase() == 'a'|| apellidoPaterno[0].toLowerCase() == 'e'|| apellidoPaterno[0].toLowerCase() == 'i'|| apellidoPaterno[0].toLowerCase()
    segundaConsonatePat14 = consonantePaterno[0].toString().toUpperCase()
}
else{
    segundaConsonatePat14 = consonantePaterno[1].toString().toUpperCase()
}
```

Aquí obtenemos el atributo de texto en la parte gráfica de la aplicación de la etiqueta apellido materno y con eso creamos la variable apellido materno que esta misma será enviada a la función sin vocal y nos devolverá una cadena solamente con consonantes y así crearemos la variable consonante materno ahora la variable apellido materno ser aprobada para ver si empieza con una vocal si es así devolveremos la primera posición de la variable consonante materno a lograr lo que creamos para este fin segunda consonante materno si no empieza con vocal devolvería una segunda posición de la variable consonante materna y por último tenemos la variable fecha la cual obtiene el atributo de texto de etiqueta en la parte gráfica de la pista de fecha

```
val apellidoMaterno = txtApeMat.text
val consonanteMaterno = SinVocal(apellidoMaterno.toString())
var segundaConsonateMat15 = ""
if (apellidoMaterno[0].toLowerCase() == 'a'|| apellidoMaterno[0].toLowerCase() == 'e'|| apellidoMaterno[0].toLowerCase() == 'i'|| apellidoMaterno[0].toLowerCase()
    segundaConsonateMat15 = consonanteMaterno[0].toString().toUpperCase()
}
else{
    segundaConsonateMat15 = consonanteMaterno[1].toString().toUpperCase()
}

val fecha = txtViewFecha.text.toString()
```

Explicación breve de cómo se resuelve

Aquí comprobamos que en la parte gráfica donde nos pide el nombre de los apellidos no estén vacíos si ese es el caso los devolverá una alerta de tipo toast de que no ha ingresado todos sus datos por el contrario cuando todo este lleno obtendremos la posición de cada valor de cada variable de arreglo en el caso

1 tenemos que pide la primera letra del apellido paterno

3 posición la primera letra del apellido materno

4 posición la primera letra del nombre

la posición 5 y 6 son las constituimos del del año

la posición 7 y 8 para el mes

la posición 9 y 10 es para el día

la posición 11 es para el género que puede ser H o M

para la posición 12 y 13 es la abertura del estado que escogimos

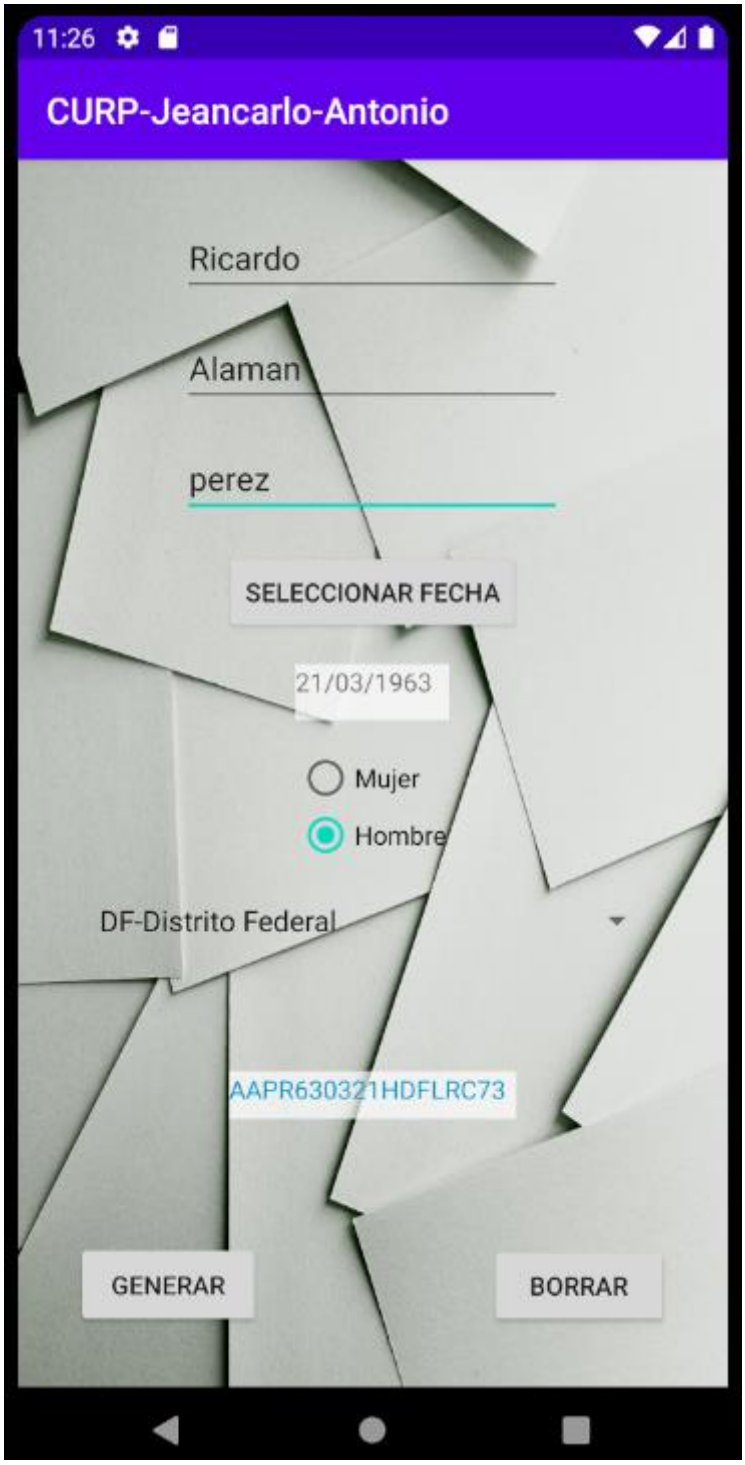
finalmente, para él dígito homo clave y el dígito verificador mandamos a llamar al arreglo de los números naturales, escogemos alguno de esos al azar

por último, asignamos a etiquetas de texto en la parte gráfica de la aplicación el atributo de tipo texto el valor ya formado de todas las posiciones de arreglo de las variables que creamos

```
if(name.length == 0 || apellidoPaterno.length == 0 || apellidoMaterno.length == 0){
    Toast.makeText( context: this, text: "No ah ingresado todos sus datos",Toast.LENGTH_SHORT).show()
}
else{
    var primerPat1 = apellidoPaterno[0].toString().toUpperCase()
    var primerMat3 = apellidoMaterno[0].toString().toUpperCase()
    var primerNombre4 = name[0].toString().toUpperCase()
    var decada5 = fecha[8]
    var año6 = fecha[9]
    var mes7 = fecha[3]
    var mes8 = fecha[4]
    var dia9 = fecha[0]
    var dia10 = fecha[1]
    var genero11 = genero
    var estado12 = txtViewEstados.text[0]
    var estado13 = txtViewEstados.text[1]
    val valor17 = mutableValues.random()
    val valor18 = mutableValues.random()

    txtViewCURP.text = primerPat1 + primerVocalPat2 + primerMat3 + primerNombre4 + decada5 + año6 + mes7
}
```

Resultado



Video Curp

https://chetumaltecnm.sharepoint.com/sites/Videos/Documentos%20compartidos/Forms/AllItems.aspx?id=%2Fsites%2FVideos%2FDocumentos%20compartidos%2FGeneral%2FRecordings%2FReuni%C3%B3n%20en%20%5FGeneral%5F%2D20210712%5F193538%2DGrabaci%C3%B3n%20de%20la%20reuni%C3%B3n%2Emp4&parent=%2Fsites%2FVideos%2FDocumentos%20compartidos%2FGeneral%2FRecordings

Repositorio Curp

https://github.com/antonio-aranda7/CURP-Jeancarlo-Antonio.git

Conclusión

Ese ejercicio de la curp es un desafío ya que tenemos que pedir diferentes tipos de datos del usuario para posteriormente procesarlos o pasarlos a tipo texto ya sea fechas y para eso utilicé los arreglos pero la complejidad está en el en el caso cuando los nombres o apellidos comienza con una vocal ya que en el curso generalmente pide la primera letra generalmente es una consonante pero sí es una vocal la tomaré igual de esta manera la primera letra el problema viene cuando después tiene que tomar ya sea la segunda buscar la segunda consonante para esto enfrentamos el reto de utilizar los arreglos extraerlos de los campos de texto y eliminar las consonantes o las vocales y probar de esta manera sí la primera del nombre apellido fue una vocal una consonante elegiremos tomar en su debido caso la segunda consonante o la segunda vocal