



**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**

**Tarea Amigo Secreto**

**NOMBRE DEL PROFESOR: Isaías May Canche**

**ALUMNO: Aranda Cen Jeancarlo Antonio**

**No. CONTROL: 17390318**

**GRUPO: I8U**

**13 DE JULIO DEL 2021**

**MARZO-JULIO 2021**

## Índice

Tarea Amigo Secreto .....	1
Índice .....	2
Instrucciones .....	3
Introducción.....	3
Detalles .....	3
Cómo se hace el RFC: .....	3
RFC.....	4
Vista del usuario .....	4
Componentes de la parte grafica.....	4
Código.....	5
Explicación de funcionamiento de las líneas de código .....	8
Comentarios acerca de la operación de la creación del curp .....	10
Explicación breve de cómo se resuelve .....	11
Resultado .....	12
Video RFC .....	12
Repositorio RFC .....	13
Conclusión .....	13

## Instrucciones

Realizar programa en Android estudio para calcular el RFC

## Introducción

En este proyecto se realizará una aplicación móvil en Android Studio con el fin de obtener datos del usuario para armar el rfc la cual es única para cada individuo el país y por si llegara a ser el caso 3 últimos dígitos especiales homoclave los cuales pueden alternarse para evitar la duplicación de la rfc

## Detalles

### *Cómo se hace el RFC:*

Los primeros que componen la clave corresponden (por lo general) al apellido paterno.

Se conforma por la primera letra del apellido y la primera vocal de este.

El tercero a la primera letra del apellido materno.

El cuarto componente es correspondiente al primer nombre.

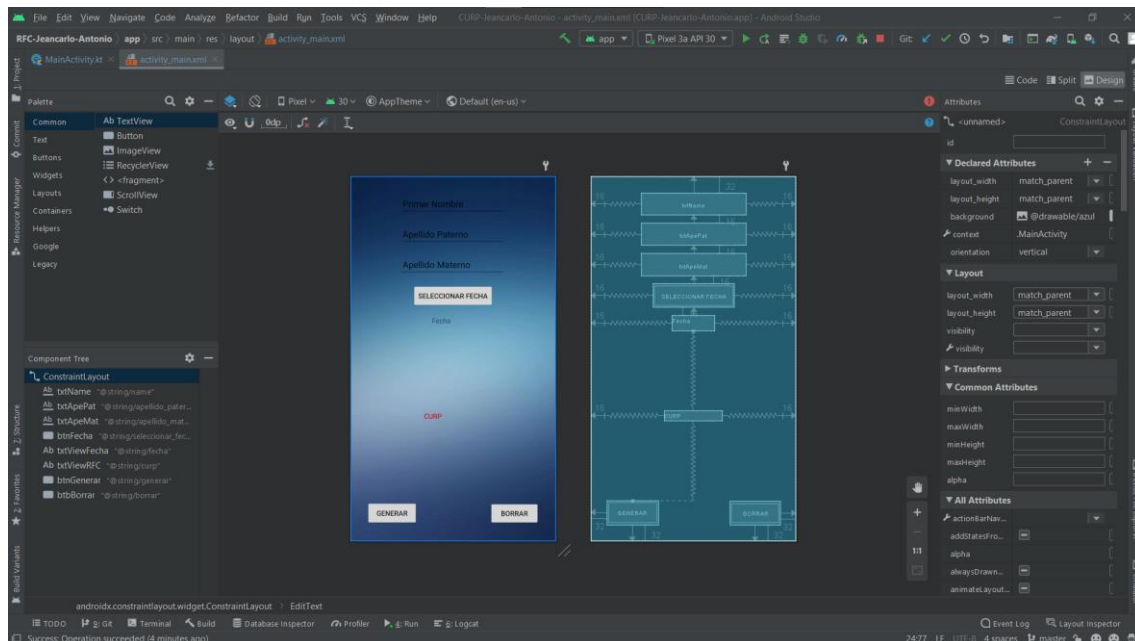
Los seis caracteres que le siguen están conformados por el año de nacimiento (0,0), mes (0,0) y día (0,0).

- Los tres dígitos últimos son una homoclave la cual es asignada por el Servicio de Administración Tributaria (SAT), para evitar las claves duplicadas.

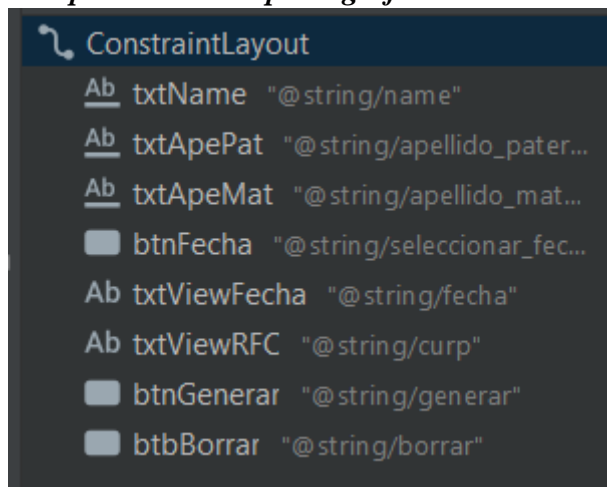


## RFC

### 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 depicker para elegir a de manera de manera más eficiente y por último un campo de texto que sólo es demostrativo en el cual nos servirá para mostrar la RFC resultante y sus debidos botones de general y borrar

## Código

```
package com.example.rfc_jeancarlo_antonio

import android.app.DatePickerDialog
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
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 ->
                        var mes = (mm + 1).toString()
                        if (mes.toInt() < 10) {
                            mes = "0" + mes
                        }
                        var dia = (dd).toString()
                        if (dia.toInt() < 10) {
                            dia = "0" + dia
                        }
                        txtViewFecha.setText("" + dia + "/" + (mes) + "/" + aaaa)
                    }, AAAA, MM, DD)
            datePickD.show()
        }

        val mutableValues = mutableListOf("0", "1", "2", "3", "4", "5", "6",
            "7", "8", "9", "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L",
            "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z"
        )

        btnGenerar.setOnClickListener {
            val name = txtName.text;

            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 apellidoMaterno = txtApeMat.text

    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]
        val valor11 = mutableValues.random()
        val valor12 = mutableValues.random()
        val valor13 = mutableValues.random()

        txtViewRFC.text =
            primerPat1 + primerVocalPat2 + primerMat3 + primerNombre4
+ decada5 + año6 + mes7 + mes8 + dia9 + dia10 + valor11 + valor12 + valor13
    }
}

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 rfc = findViewById<TextView>(R.id.txtViewRFC)
    rfc.setText("RFC")
}

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

//*****
*****->
    //Metodo para eliminar las consonantes de los textos Nombre y Apellidos
    fun SinConsonante( text: String ): String {
        //Contruye una cadena
        val resultado = StringBuilder()
        //Recorre la cadena si hay texto
        for (char in text) {
            //Si son vocales las guarda en el arreglo
            if (!"bcdfghjklmnñpqrstvwxyz".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 datePicker 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 con 1 ya que empiezan desde cero y la convertimos a tipo texto i por consiguiente como la curva 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 i 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 datePickerD = 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)
        datePickerD.show()
    }
}
```

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

```
val mutableValues = mutableListOf("0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z")
```

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, al de fecha y rfc les asignará estos 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 rfc = findViewById<TextView>(R.id.txtViewRFC)
    rfc.setText("RFC")
}

```

Por último, tenemos una función en este caso le nombramos sin consonante de acuerdo obtendrá un arreglo 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 y si encuentra alguna letra consonante las eliminará y sólo nos quedarán las vocales y este último arreglo lo devolverá

```

//Metodo para eliminar las consonantes de los textos Nombre y Apellidos
fun SinConsonante( text: String ): String {
    //Contruye una cadena
    val resultado = StringBuilder()
    //Recorre la cadena si hay texto
    for (char in text) {
        //Si son vocales las guarda en el arreglo
        if (!"bcdfghjklmnpqrstvwxyz".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 apellidoPaterno = txtApePat.text
    var apellidoVocal = SinConsonante(apellidoPaterno.toString())
    var primerVocalPat2 = ""

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

    val apellidoMaterno = txtApeMat.text

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

El manejador del evento del botón general al ser presionado creará una oleada de nombre la cual se asociará el primer nombre que hemos ingresado lo mismo para el apellido paterno el apellido materno y la fecha tenemos el detepicker la transforma a una cadena.

También tenemos que sobre ellos en variable apellido vocal es la variable apellido paterno, pero sólo hecha con vocales

Entonces ahora pregunta si el apellido paterno en la primera posición este empieza con una vocal si es así para el siguiente espacio que sería la variable dos en este caso primera vocal las dos se reinaría con la con la segunda vocal del apellido paterno si no pues sería la primera vocal que encuentre

### ***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 todas 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

para la posición 11, 12 y 13 es finalmente la homoclave donde mandamos a llamar al arreglo de los números naturales y letras del alfabeto, 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]  
    val valor11 = mutableValues.random()  
    val valor12 = mutableValues.random()  
    val valor13 = mutableValues.random()  
  
    textViewRFC.text =  
        primerPat1 + primerVocalPat2 + primerMat3 + primerNombre4 + decada5 + año6 + mes7 + mes8 + dia9 + dia10 + valor11 + valor12 + valor13  
}
```

### **Resultado**

The screenshot shows a mobile application interface with a purple header bar containing the text "RFC-Jeancarlo-Antonio". Below the header, there are three input fields with labels "Primer Nombre", "Apellido Paterno", and "Apellido Materno". A button labeled "SELECCIONAR FECHA" is positioned above a date input field labeled "Fecha". Below the date field is an input field labeled "RFC" in red text. At the bottom of the screen, there are two buttons: "GENERAR" and "BORRAR". The status bar at the top shows the time as 9:04 and various icons.

### **Video RFC**

<https://chetumaltecnm.sharepoint.com/sites/Videos/Documentos%20compartidos/Forms/AllItems.aspx?id=%2Fsites%2FVideos%2FDocumentos%20compartidos%2FGeneral%2FRecordings%2FNueva%20reuni%C3%B3n%20de%20canal%2D20210713%5F164512%2DGrabaci%C3%B3n%20de%20la%20reuni%C3%B3n%2Emp4&parent=%2Fsites%2FVideos%2FDocumentos%20compartidos%2FGeneral%2FRecordings>

### ***Repositorio RFC***

<https://github.com/antonio-aranda7/RFC-Jeancarlo-Antonio.git>

### **Conclusión**

Este ejercicio fue un desafío considerable de que tenemos que pedir distintos datos del usuario procesarlos i para eso se utilizaron arreglos y las posiciones de estos para el final crear una cadena que es la que se mostraría como respuesta