



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



QUINTANA ROO
GOBIERNO DEL ESTADO
2016-2022



SEQ
SECRETARÍA DE EDUCACIÓN

Instituto Tecnológico de Chetumal

Programación de Dispositivos Móviles I

Ingeniería en sistemas computacionales I8U



 Sheila Elizabeth Vázquez Chimay



TECNOLÓGICO
NACIONAL DE MÉXICO

RFC

```
package com.example.examenrfc

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Spinner
import android.widget.AdapterView
import android.widget.TextView
import kotlinx.android.synthetic.main.activity_main.*
import java.util.ArrayList

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

        //creación de spinner para meses del año
        val spinnerMes = findViewById<Spinner>(R.id.spinnerMes)
        val valueMes =
            arrayOf("01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12")
        val arrayMes =
            ArrayAdapter<String>(this, android.R.layout.simple_spinner_item, valueMes)
        spinnerMes.adapter = arrayMes

        //creación de spinner para los días del mes.
        val spinnerDia = findViewById<Spinner>(R.id.spinnerDia)
        val valueDia =
            arrayOf("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31")
        val arrayDia =
            ArrayAdapter<String>(this, android.R.layout.simple_spinner_item, valueDia)
        spinnerDia.adapter = arrayDia

        //creación de spinner para años del 1950 a 2020
        val year = ArrayList<String>()
        for (i in 1950 until 2020){
            year.add(i.toString())
        }
        val spinnerYear = findViewById<Spinner>(R.id.spinnerYear)
        val arrayYear =
            ArrayAdapter<String>(this, android.R.layout.simple_spinner_item, year)
        spinnerYear.adapter = arrayYear
    }

    //creación de función para generar el RFC
    fun generar(view: View) {
        //Recibimos los valores guardados y seleccionados en los txt y spinners
        val nombreValue = txtNombre.text.toString()
        val apellidopValue = txtApellidop.text.toString()
        val apellidomValue = txtApellidom.text.toString()
        val diaValue = spinnerDia.selectedItem.toString()
        val mesValue = spinnerMes.selectedItem.toString()
        val yearValue = spinnerYear.selectedItem.toString()
    }
}
```

```

        val arrayApep = apellidoValue.split("")
        val arrayApem = apellidoValue.split("")
        val arrayNombre = nombreValue.split("")
        val arrayYear = yearValue.split("")

        //creación de la homoclave. Podrá ser del 1 al 9 y de la A a la Z, la
        homoclave son tres números aleatorios.
        val homoclave =
arrayOf("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")
        val homo1 = homoclave.random()
        val homo2 = homoclave.random()
        val homo3 = homoclave.random()

        //Para conseguir la segunda vocal haremos uso de find que devuelve el
        primer elemento que coincide con los predicados dado. En este caso del array del
        apellido paterno le decimos que encuentre A,E,I,O,U.
        val segundaVocal =
arrayApep.find{it.equals("a")||it.equals("e")||it.equals("i")||it.equals("o")||i
t.equals("u")})
        //La vocal obtenida la convertimos a Mayúsculas con toUpperCase()
        val segundavocal = segundaVocal.toString().toUpperCase()

        // Imprimimos el RFC en el orden solicitado, la posición 1 del array
        donde se encuentra el apellido paterno, la segunda vocal del apellido paterno,
        la posición 1 del apellido materno, la primera letra del nombre, la posición 3 y
        4 del año es decir, si es 1999 se mostrara 99, el mes, día y los tres números
        aleatorios que forman la homoclave.
        var rfc = findViewById<TextView>(R.id.txtRFC)
        rfc.text = arrayApep[1].toUpperCase() + segundavocal +
arrayApem[1].toUpperCase() + arrayNombre[1].toUpperCase() + arrayYear[3]+
arrayYear[4] + mesValue + diaValue + homo1 + homo2 + homo3
    }

función para el boton de limpiar, es decir dejamos vacíos los txt ,los spinner
regresan a la posición 0 y borramos el rfc anterior.
    fun limpiar(view: View) {
        txtNombre.text.clear()
        txtApellidoop.text.clear()
        txtApellidoom.text.clear()
        spinnerDia.setSelection(0)
        spinnerMes.setSelection(0)
        spinnerYear.setSelection(0)
        txtRFC.text = ""
    }
}

```

CURP

```
package com.example.examencurp

//Se agregan las librerías requeridas
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.AdapterView
import android.widget.AdapterView.OnItemClickListener
import android.widget.ArrayAdapter
import android.widget.Spinner
import android.widget.TextView
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)

        //Creamos un spinner con el sexo de Mujer o Hombre
        val spinnerSexo = findViewById<Spinner>(R.id.spinnerSexo)
        val valueSexo = arrayOf("Hombre", "Mujer")
        val arraySexo =
            ArrayAdapter(this, android.R.layout.simple_spinner_item, valueSexo)
        spinnerSexo.adapter = arraySexo

        //Creamos un spinner con los Estados del país.
        val spinnerEstado = findViewById<Spinner>(R.id.spinnerEstado)
        val valueEstado = arrayOf("AS-Aguascalientes", "BC-Baja California",
            "BS-Baja California Sur",
            "CC-Campeche", "CS-Chiapas", "CH-Chihuahua", "DF-Ciudad de México",
            "CL-Coahuila",
            "CM-Colima", "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", "QO-Querétaro", "QR-Quintana Roo", "SP-San Luis
Potosí", "SL-Sinaloa",
            "SR-Sonora", "TC-Tabasco", "TS-Tamaulipas", "TL-Tlaxcala", "VZ-
Veracruz", "YN-Yucatan", "ZS-Zacatecas")
        val arrayEstado =
            ArrayAdapter(this, android.R.layout.simple_spinner_item, valueEstado)
        spinnerEstado.adapter = arrayEstado

        //creación de spinner para los meses del año.
        val spinnerMes = findViewById<Spinner>(R.id.spinnerMes)
        val valueMes =
            arrayOf("01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12")
        val arrayMes =
            ArrayAdapter(this, android.R.layout.simple_spinner_item, valueMes)
        spinnerMes.adapter = arrayMes

        //creación de spinner para los días del mes.
        val spinnerDia = findViewById<Spinner>(R.id.spinnerDia)
        val valueDia =
            arrayOf("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "
```

```

17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31")
    val arrayDia =
        ArrayAdapter(this, android.R.layout.simple_spinner_item, valueDia)
        spinnerDia.adapter = arrayDia

    //creación de spinner para los años de 1950 hasta 2020.
    val year = ArrayList<String>()
    for (i in 1950 until 2020){
        year.add(i.toString())
    }
    val spinnerYear = findViewById<Spinner>(R.id.spinnerYear)
    val arrayYear =
        ArrayAdapter(this, android.R.layout.simple_spinner_item, year)
        spinnerYear.adapter = arrayYear
    }

//función para generar la CURP
fun generar(view: View) {
    //Recibimos los valores guardados y seleccionados en los txt y spinners
    val nombreValue = txtNombre.text.toString()
    val apellidopValue = txtApellidop.text.toString()
    val apellidomValue = txtApellidom.text.toString()
    val diaValue = spinnerDia.selectedItem.toString()
    val mesValue = spinnerMes.selectedItem.toString()
    val yearValue = spinnerYear.selectedItem.toString()
    val sexoValue = spinnerSexo.selectedItem.toString()
    val estadoValue = spinnerEstado.selectedItem.toString()

    val arrayApep = apellidopValue.split("")
    val arrayApem = apellidomValue.split("")
    val arrayNombre = nombreValue.split("")
    val arrayYear = yearValue.split("")

    //creación de la homoclave. Podrá ser del 1 al 9 y de la A a la Z, la
    homoclave son tres números aleatorios.
    val homoclave =
        arrayOf("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")
    val homo1 = homoclave.random()
    val homo2 = homoclave.random()
    val homo3 = homoclave.random()

    //Para conseguir la segunda vocal haremos uso de find que devuelve el
    primer elemento que coincide con los predicados dado. En este caso del array del
    apellido paterno le decimos que encuentre A,E,I,O,U.
    val segundaVocal =
        arrayApep.find{it.equals("a")||it.equals("e")||it.equals("i")||it.equals("o")||i
            t.equals("u")}
    //La vocal obtenida la convertimos a Mayúsculas con toUpperCase()
    val segundavocal = segundaVocal.toString().toUpperCase()

```

```

//De nombreValue eliminaremos la primera letra con ayuda de Drop y
separaremos la cadena con Split. Después buscaremos en la nueva variable la
segunda de las siguientes consonantes del abecedario que aparezcan en el nombre.
val arrayName = nombreValue.drop(1).split("")
val consonanteName = arrayName.find
{it.equals("b")||it.equals("c")||it.equals("d")||it.equals("f")

||it.equals("g")||it.equals("h")||it.equals("j")||it.equals("k")||it.equals("l")

||it.equals("m")||it.equals("n")||it.equals("ñ")||it.equals("p")||it.equals("q")

||it.equals("r")||it.equals("s")||it.equals("t")||it.equals("v")||it.equals("w")
||it.equals("x")||it.equals("y")||it.equals("z")}
//Convertiremos el resultado de la búsqueda en mayúsculas.
var consonanteNombre = consonanteName.toString().toUpperCase()

//De apellidopValue eliminaremos la primera letra con ayuda de Drop y
separaremos la cadena con Split. Después buscaremos en la nueva variable la
segunda de las siguientes consonantes del abecedario que aparezcan en el
apellido parteno.
val arrayApellidoP = apellidopValue.drop(1).split("")
var consonanteApeP = arrayApellidoP.find
{it.equals("b")||it.equals("c")||it.equals("d")||it.equals("f")

||it.equals("g")||it.equals("h")||it.equals("j")||it.equals("k")||it.equals("l")

||it.equals("m")||it.equals("n")||it.equals("ñ")||it.equals("p")||it.equals("q")

||it.equals("r")||it.equals("s")||it.equals("t")||it.equals("v")||it.equals("w")
||it.equals("x")||it.equals("y")||it.equals("z")}
//Convertiremos el resultado de la búsqueda en mayúsculas.
var consonantePaterno = consonanteApeP.toString().toUpperCase()

//Haremos lo mismo otra vez que con el nombre y el apellido paterno solo
que ahora con el apellido materno.
val arrayApellidoM = apellidomValue.drop(1).split("")
var consonanteApeM = arrayApellidoM.find
{it.equals("b")||it.equals("c")||it.equals("d")||it.equals("f")

||it.equals("g")||it.equals("h")||it.equals("j")||it.equals("k")||it.equals("l")

||it.equals("m")||it.equals("n")||it.equals("ñ")||it.equals("p")||it.equals("q")

||it.equals("r")||it.equals("s")||it.equals("t")||it.equals("v")||it.equals("w")
||it.equals("x")||it.equals("y")||it.equals("z")}
//Convertiremos el resultado de la búsqueda en mayúsculas.
var consonanteMaterno = consonanteApeM.toString().toUpperCase()

//Para armar la CURP haremos uso de las posiciones y de los valores
sacador anteriormente. En la primera posición indicaremos la primera letra del
apellido paterno, como siguiente valor será la primera vocal del apellido
paterno, en tercera posición colocaremos la primera letra del apellido materno,
en cuarta posición colocaremos la primera letra del nombre, la posición 3 y 4
del año es decir, si es 1999 se mostrara 99, el mes, día, el sexo es decir "M" o
"H", el estado que contiene las abreviaturas, la consonante del apellido

```

paterno, la consonante del apellido materno, la consonante del nombre y los tres números aleatorios que forman la homoclave.

```
var curp = findViewById<TextView>(R.id.txtRFC)
curp.text = arrayApep[1].toUpperCase() + segundavocal +
arrayApem[1].toUpperCase() + arrayNombre[1].toUpperCase() + arrayYear[3]+
arrayYear[4] +
        mesValue + diaValue + sexoValue[0] + estadoValue[0] +
estadoValue[1] + consonantePaterno + consonanteMaterno + consonanteNombre +homo1
+ homo2 + homo3
}
```

función para el boton de limpiar, es decir dejamos vacíos los txt ,los spinner regresan a la posición 0 y borramos el curp anterior.

```
fun limpiar(view: View) {
    txtNombre.text.clear()
    txtApellidop.text.clear()
    txtApellidom.text.clear()
    spinnerDia.setSelection(0)
    spinnerMes.setSelection(0)
    spinnerYear.setSelection(0)
    spinnerSexo.setSelection(0)
    spinnerEstado.setSelection(0)
    txtRFC.text = ""
}
}
```