



UNIVERSIDAD DON BOSCO  
FACULTAD DE INGENIERIA  
ESCUELA DE INGENIERIA EN COMPUTACIÓN  
DESARROLLO DE SOFTWARE PARA MÓVILES

Machado Rivas, Mauricio Alexander MR171225

Grupo. 03L

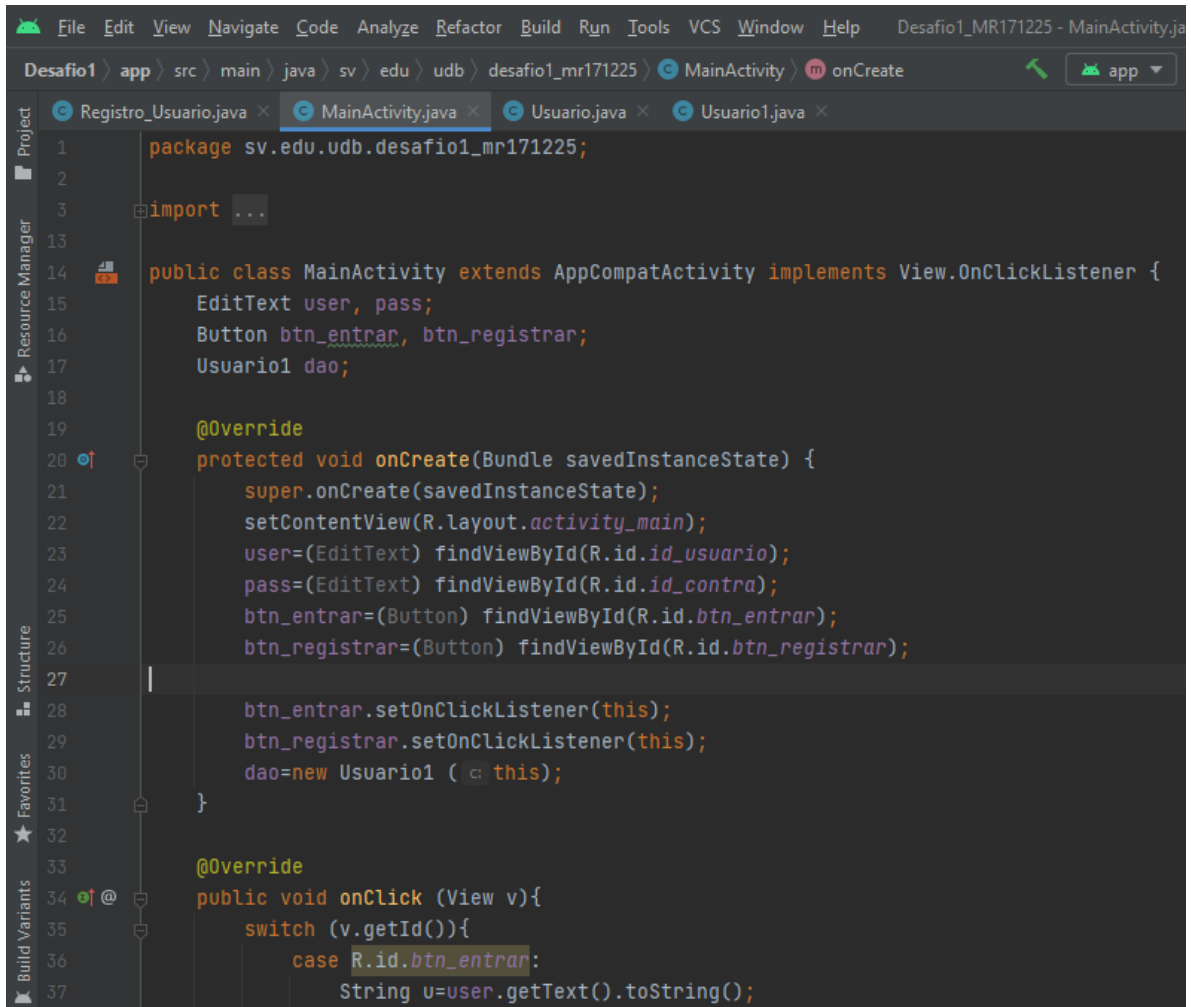
Docente. Ing. Alexander Alberto Sigüenza

Actividad. 1er. Desafío Práctico

Soyapango 13, Marzo 2022

Login -----→100%

Codificación, Usuario ya registrado



```
Desafio1_MR171225 - MainActivity.java
Desafio1 > app > src > main > java > sv > edu > udb > desafio1_mr171225 > MainActivity > onCreate
Registro_Usuario.java x MainActivity.java x Usuario.java x Usuario1.java x
1 package sv.edu.udb.desafio1_mr171225;
2
3 import ...
13
14 public class MainActivity extends AppCompatActivity implements View.OnClickListener {
15     EditText user, pass;
16     Button btn_entrar, btn_registrar;
17     Usuario1 dao;
18
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_main);
23         user=(EditText) findViewById(R.id.id_usuario);
24         pass=(EditText) findViewById(R.id.id_contra);
25         btn_entrar=(Button) findViewById(R.id.btn_entrar);
26         btn_registrar=(Button) findViewById(R.id.btn_registrar);
27
28         btn_entrar.setOnClickListener(this);
29         btn_registrar.setOnClickListener(this);
30         dao=new Usuario1 ( this);
31     }
32
33     @Override
34     public void onClick (View v){
35         switch (v.getId()){
36             case R.id.btn_entrar:
37                 String u=user.getText().toString();
```

```
@Override
public void onClick (View v){
    switch (v.getId()){
        case R.id.btn_entrar:
            String u=user.getText().toString();
            String p=pass.getText().toString();
            if (u.equals("")&&p.equals("")){
                Toast.makeText( context: this, text: "ERROR: Campos Vacios",Toast.LENGTH_LONG).show();
            } else if (dao.Login(u,p)!=1){
                Usuario ux=dao.getUsuario(u,p);
                Toast.makeText( context: this, text: "Datos Correctos", Toast.LENGTH_LONG).show();
                Intent i2=new Intent( packageContext: MainActivity.this, Menu.class);
                startActivity(i2);
                finish();
            } else{
                Toast.makeText( context: this, text: "Usuario y/o Contraseña Incorrectos",Toast.LENGTH_LONG).show();
            }

            break;
        case R.id.btn_registrar:
            Intent i=new Intent( packageContext: MainActivity.this, Registro_Usuario.class);
            startActivity(i);
            break;
    }
}
```

Usuario no Registrado

Clase que contiene la base de datos

```
package sv.edu.udb.desafio1_mr171225;

import ...

public class Usuario1 {
    Context c;
    Usuario u;
    ArrayList<Usuario> lista;
    SQLiteDatabase sql;
    String bd="BDDesafio1";
    String tabla="create table if not exists usuario (id integer primary key autoincrement, usuario text, contra text, nombre text, apellido text)";

    public Usuario1 (Context c){
        this.c=c;
        sql= c.openOrCreateDatabase(bd, c.MODE_PRIVATE, factory: null);
        sql.execSQL(tabla);
        u=new Usuario();
    }

    public boolean insertarUsuario (Usuario u){

        ContentValues cv=new ContentValues();
        cv.put("usuario", u.getUsuario());
        cv.put("contra", u.getContraseña());
        cv.put("nombre", u.getNombre());
        cv.put("apellido", u.getApellidos());
        return (sql.insert( table: "usuario", nullColumnHack: null, cv)>0);
    }
}
```

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Desafio1_MR171225 - Usuario1.java [Desafio1_MR171225.app]
Desafio1 / app / src / main / java / sv / edu / udb / desafio1_mr171225 / Usuario1
Registro_Usuario.java x MainActivity.java x Usuario.java x Usuario1.java x

36
37 public ArrayList<Usuario>selectUsuarios(){
38     ArrayList<Usuario>lista=new ArrayList<>();
39     lista.clear();
40     Cursor cr=sql.rawQuery( sql: "select * from usuario", selectionArgs: null);
41     if (cr!=null &&cr.moveToFirst()){
42         do{
43             Usuario u=new Usuario();
44             u.setId(cr.getInt( columnIndex: 0));
45             u.setUsuario(cr.getString( columnIndex: 1));
46             u.setContraseña(cr.getString( columnIndex: 2));
47             u.setNombre(cr.getString( columnIndex: 3));
48             u.setApellidos(cr.getString( columnIndex: 4));
49             lista.add(u);
50         }while (cr.moveToNext());
51     }
52     return lista;
53 }
54
55 public int login (String u, String p){
56     int a=0;
57     Cursor cr=sql.rawQuery( sql: "select * from usuario", selectionArgs: null);
58     if (cr!=null&&cr.moveToFirst()){
59         do {
60             if (cr.getString( columnIndex: 1).equals(u) && cr.getString( columnIndex: 2).equals(p)){
61                 a++;
62             }
63         }while (cr.moveToNext());
64     }
65 }
```

```
Registro_Usuario.java x MainActivity.java x Usuario.java x Usuario1.java x
63         }while (cr.moveToNext());
64     }
65     return a;
66 }
67
68 public Usuario getUsuario (String u, String p){
69     lista=selectUsuarios();
70     for (Usuario us:lista){
71         if (us.getUsuario().equals(u)&&us.getContraseña().equals(p)){
72             return us;
73         }
74     }
75     return null;
76 }
77
78 public Usuario getUsuarioById(int id){
79     lista=selectUsuarios();
80     for (Usuario us:lista){
81         if (us.getId()==id){
82             return us;
83         }
84     }
85     return null;
86 }
87
88 }
89
90 }
```

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Desafio1_MR171225 - Usuario.java [Desafio1_MR171225.app]
Desafio1 > app > src > main > java > sv > edu > udb > desafio1_mr171225 > Usuario
Registro_Usuario.java x MainActivity.java x Usuario.java x Usuario1.java x
Project
Resource Manager
Structure
Favorites
25
1 package sv.edu.udb.desafio1_mr171225;
2
3 public class Usuario {
4     int id; String Nombre, Apellidos, Usuario, Contraseña;
5
6     public Usuario(){
7
8     }
9
10    public Usuario(int id, String nombre, String apellidos, String usuario, String contraseña) {
11        this.id = id;
12        Nombre = nombre;
13        Apellidos = apellidos;
14        Usuario = usuario;
15        Contraseña = contraseña;
16    }
17
18    public boolean isNull() {
19        if (Nombre.equals("") && Apellidos.equals("") && Usuario.equals("") && Contraseña.equals("")) {
20            return false;
21        } else {
22            return true;
23        }
24    }
25
```

```
Registro_Usuario.java x MainActivity.java x Usuario.java x Usuario1.java x
Project
Resource Manager
Structure
Favorites
Build Variants
25
26 @Override
27 public String toString() {
28     return "Usuario{" +
29         "id=" + id +
30         ", Nombre='" + Nombre + '\'' +
31         ", Apellidos='" + Apellidos + '\'' +
32         ", Usuario='" + Usuario + '\'' +
33         ", Contraseña='" + Contraseña + '\'' +
34         '}';
35 }
36
37 public int getId() { return id; }
40
41 public void setId(int id) { this.id = id; }
44
45 public String getNombre() { return Nombre; }
48
49 public void setNombre(String nombre) { Nombre = nombre; }
52
53 public String getApellidos() { return Apellidos; }
56
57 public void setApellidos(String apellidos) { Apellidos = apellidos; }
60
61 public String getUsuario() { return Usuario; }
64
65 public void setUsuario(String usuario) { Usuario = usuario; }
68
```

```

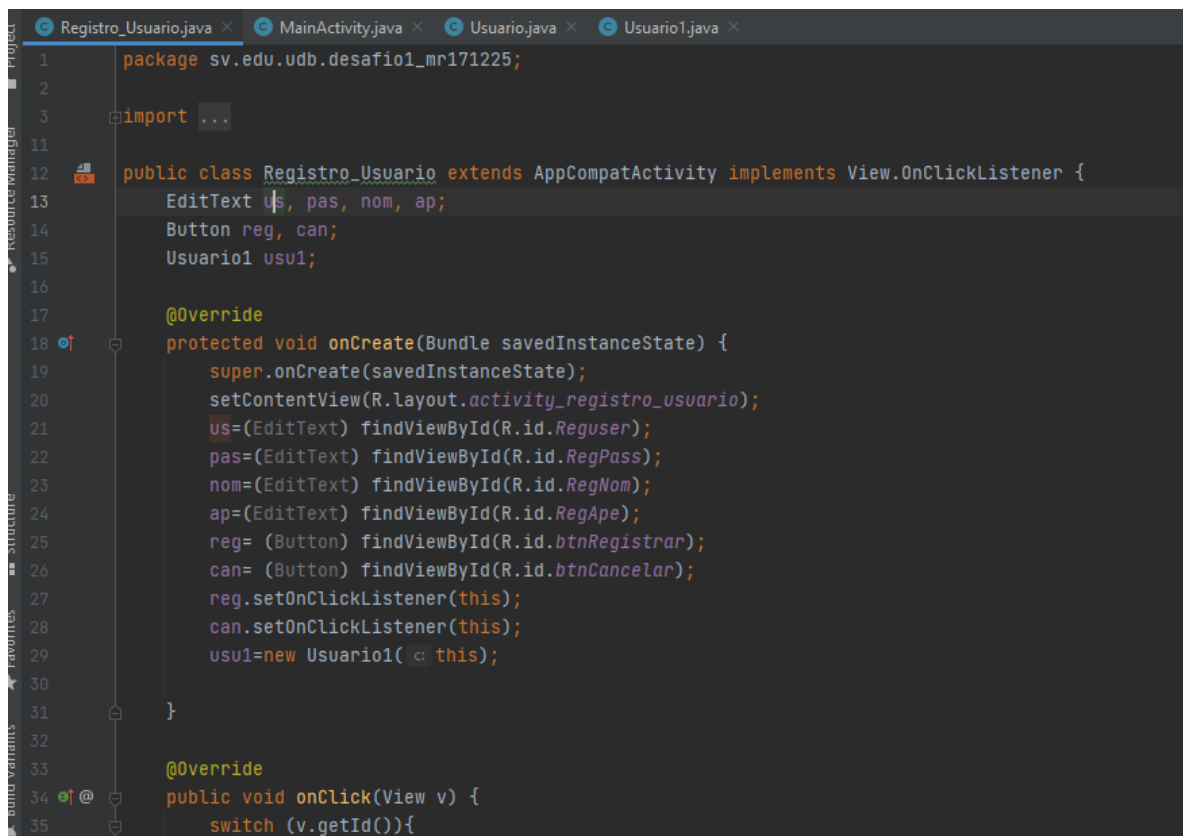
public void setUsuario(String usuario) { Usuario = usuario; }

public String getContraseña() { return Contraseña; }

public void setContraseña(String contraseña) { Contraseña = contraseña; }

```

## Registro\_Usuario.java



```

1 package sv.edu.udb.desafio1_mr171225;
2
3 import ...
4
11
12 public class Registro_Usuario extends AppCompatActivity implements View.OnClickListener {
13     EditText us, pas, nom, ap;
14     Button reg, can;
15     Usuario1 usu1;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_registro_usuario);
21         us=(EditText) findViewById(R.id.Reguser);
22         pas=(EditText) findViewById(R.id.RegPass);
23         nom=(EditText) findViewById(R.id.RegNom);
24         ap=(EditText) findViewById(R.id.RegApe);
25         reg= (Button) findViewById(R.id.btnRegistrar);
26         can= (Button) findViewById(R.id.btnCancelar);
27         reg.setOnClickListener(this);
28         can.setOnClickListener(this);
29         usu1=new Usuario1( this);
30
31     }
32
33     @Override
34     public void onClick(View v) {
35         switch (v.getId()){

```

```

@Override
public void onClick(View v) {
    switch (v.getId()){
        case R.id.btnRegistrar:
            Usuario u=new Usuario();
            u.setUsuario(us.getText().toString());
            u.setContraseña(pas.getText().toString());
            u.setNombre(nom.getText().toString());
            u.setApellidos(ap.getText().toString());
            if (!u.isNull()){
                Toast.makeText( context: this, text: "ERROR: Campos Vacios", Toast.LENGTH_LONG).show();
            } else if (usu1.insertarUsuario(u)) {
                Toast.makeText( context: this, text: "Registro Exitoso!!!", Toast.LENGTH_LONG).show();
                Intent i2=new Intent( packageContext: Registro_Usuario.this, MainActivity.class);
                startActivity(i2);
                finish();
            }else{
                Toast.makeText( context: this, text: "Usuario ya Registrado!!!", Toast.LENGTH_LONG).show();
            }

            break;
        case R.id.btnCancelar:
            Intent i=new Intent( packageContext: Registro_Usuario.this, MainActivity.class);
            startActivity(i);
            finish();
            break;
    }
}

```

```

        break;
    case R.id.btnCancelar:
        Intent i=new Intent( packageContext: Registro_Usuario.this, MainActivity.class);
        startActivity(i);
        finish();
        break;
    }
}
}

```

## Estructura de Pantallas



3:47 100%

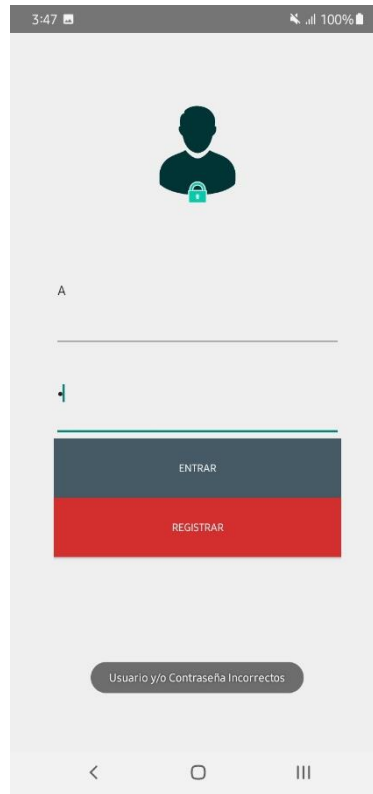


Usuario


Contraseña

ENTRAR

REGISTRAR



3:47 100%



A

1

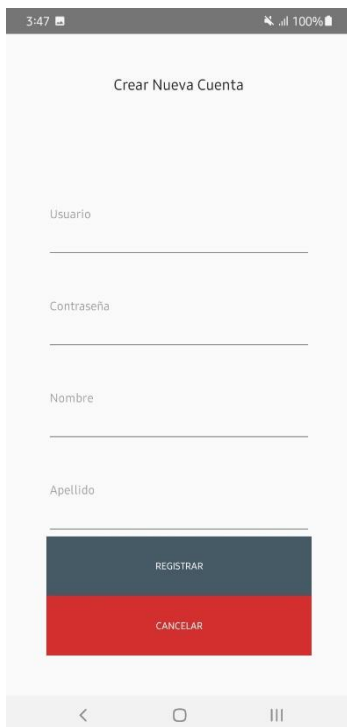
ENTRAR

REGISTRAR

Usuario y/o Contraseña Incorrectos

Usuario que no ha sido registrado aún.

## Registrar Nuevo Usuario



3:47 100%

Crear Nueva Cuenta

Usuario

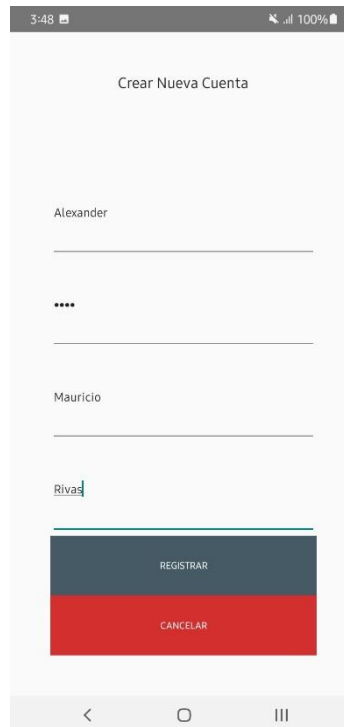
Contraseña

Nombre

Apellido

REGISTRAR

CANCELAR



3:48 100%

Crear Nueva Cuenta

Alexander

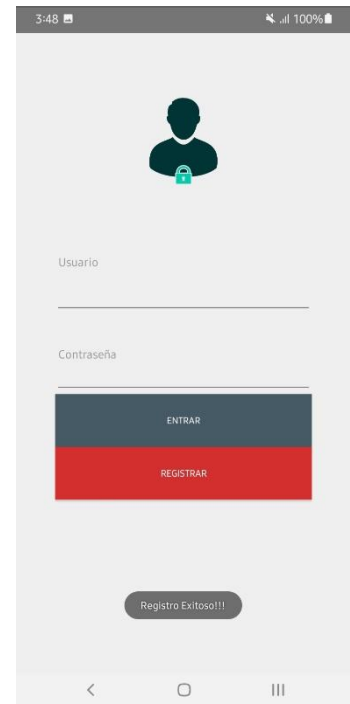
....

Mauricio


Rivas

REGISTRAR

CANCELAR



3:48 100%



Usuario

Contraseña

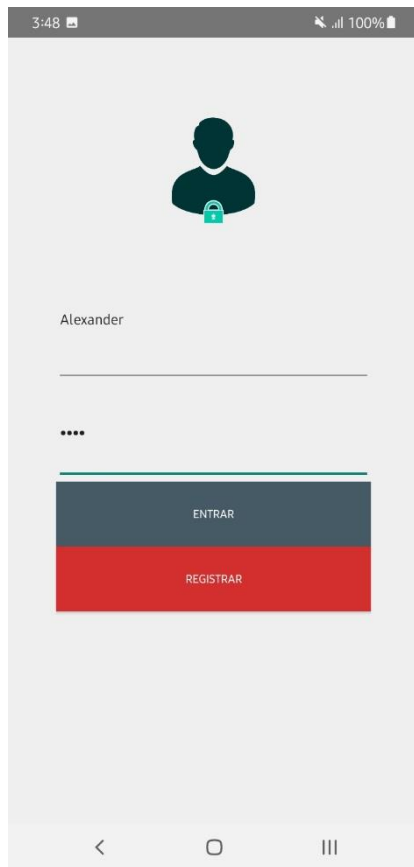
ENTRAR

REGISTRAR

Registro Exitoso!!!



## Loguearse con Usuario ya Registrado



Una vez hemos  
ingresado con el  
Usuario registrado  
nos mandará a la  
pantalla de menú



## Pantalla de Menú.



Ejercicio 1 -----→ 100%

Realizar una aplicación móvil que resuelva la solución de la ecuación cuadrática

```
Desafio1 / app / src / main / java / sv / edu / udb / desafio1_mr171225 / Ejercicio_1 / onClick
Ejercicio_1.java
1 package sv.edu.udb.desafio1_mr171225;
2
3 import ...
4
5
6
7
8
9
10
11 public class Ejercicio_1 extends AppCompatActivity implements View.OnClickListener {
12     Button bt_calcular;
13     EditText id_a, id_b, id_c;
14     TextView id_x1, id_x2, id_res;
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_ejercicio1);
20
21         bt_calcular = (Button) findViewById(R.id.btn_calcular);
22         id_a = (EditText) findViewById(R.id.id_a);
23         id_b = (EditText) findViewById(R.id.id_b);
24         id_c = (EditText) findViewById(R.id.id_c);
25         id_x1 = (TextView) findViewById(R.id.id_x1);
26         id_x2 = (TextView) findViewById(R.id.id_x2);
27         id_res = (TextView) findViewById(R.id.id_res);
28         bt_calcular.setOnClickListener(this);
29     }
30
31     @Override
32     public void onClick(View v) {
33
```

```
Ejercicio_1.java
30
31     @Override
32     public void onClick(View v) {
33
34         double a = Double.parseDouble(id_a.getText().toString());
35         double b = Double.parseDouble(id_b.getText().toString());
36         double c = Double.parseDouble(id_c.getText().toString());
37
38         double x1 = (-b + (Math.sqrt(Math.pow(b, 2) - (4 * a * c)))) / 2 * a;
39         double x2 = (-b - (Math.sqrt(Math.pow(b, 2) - (4 * a * c)))) / 2 * a;
40         double D = Math.pow(b, 2) - (4 * a * c);
41
42         id_x1.setText(x1 + "");
43         id_x2.setText(x2 + "");
44
45         if (D >= 0) {
46             id_res.setText("La ecuación tiene solución con Números Reales");
47         } else {
48             id_res.setText("La ecuación no tiene solución con Números Reales");
49         }
50     }
51 }
52
53
```

## Funcionalidad.

3:48

100%

Solución Sistema de Ecuación Cuadrática

$$ax^2 + bx + c = 0$$
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Por Favor Ingrese el Valor de Cada Incognita

a:

b:

c:

CALCULAR

1er. Solución

2da. Solución

<

○

|||

3:49

100%

Solución Sistema de Ecuación Cuadrática

$$ax^2 + bx + c = 0$$
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Por Favor Ingresa el Valor de Cada Incognita

2

9

10

CALCULAR

1er. Solución

2da. Solución

<

○

|||

3:49

100%

Solución Sistema de Ecuación Cuadrática

$$ax^2 + bx + c = 0$$
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Por Favor Ingresa el Valor de Cada Incognita

2

9

10

CALCULAR

-8.0

-10.0

La ecuación tiene solución con Números Reales

<

○

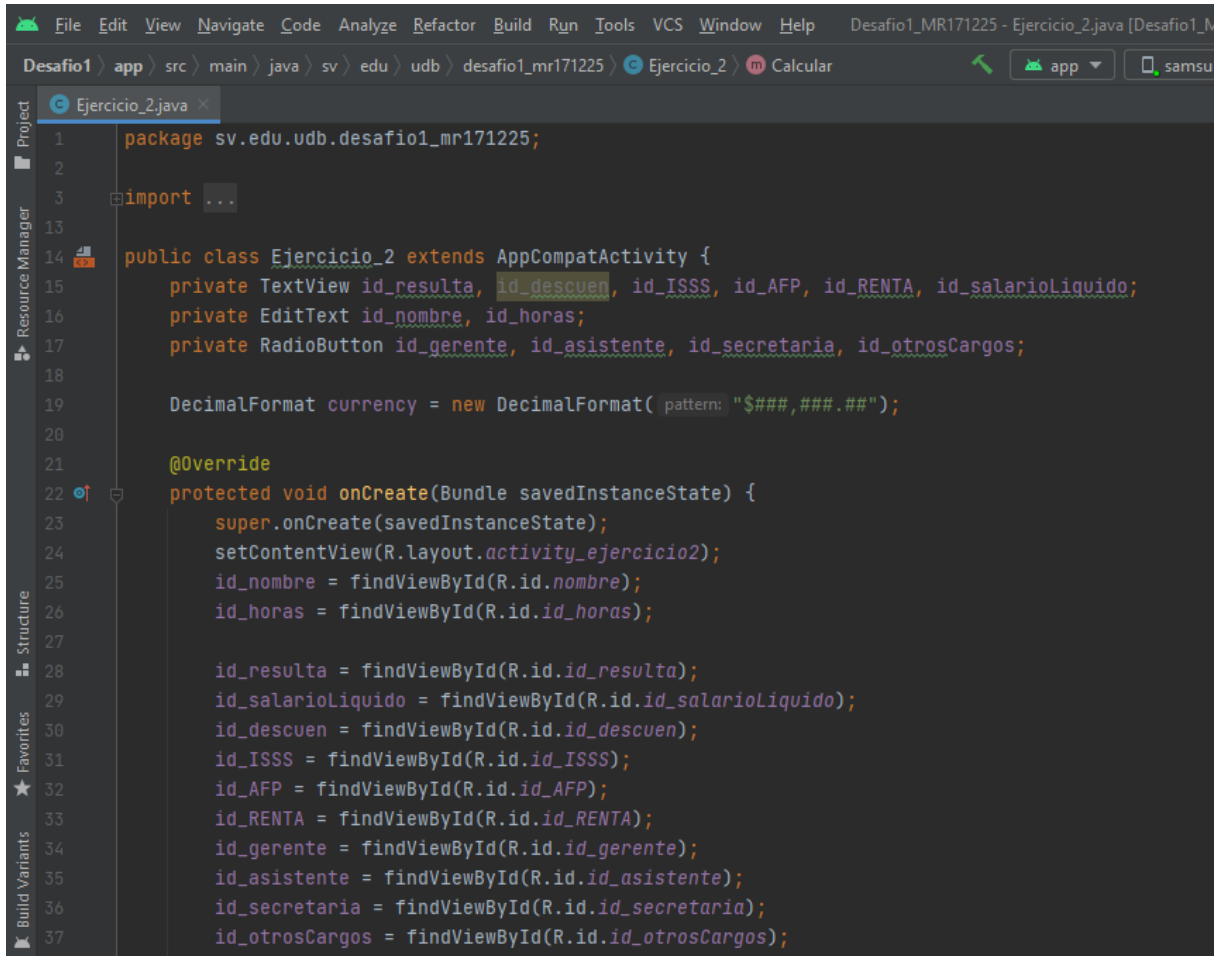
|||

Ejercicio 2 -----→0%

Ejercicio 3 -----→75%

Calcular el pago líquido de un empleado

## CODIFICACIÓN



```
1 package sv.edu.udb.desafio1_mr171225;
2
3 import ...
4
13
14 public class Ejercicio_2 extends AppCompatActivity {
15     private TextView id_resulta, id_descuen, id_ISSS, id_AFP, id_RENTA, id_salarioLiquido;
16     private EditText id_nombre, id_horas;
17     private RadioButton id_gerente, id_asistente, id_secretaria, id_otrosCargos;
18
19     DecimalFormat currency = new DecimalFormat( pattern: "$###,###.##");
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_ejercicio2);
25         id_nombre = findViewById(R.id.nombre);
26         id_horas = findViewById(R.id.id_horas);
27
28         id_resulta = findViewById(R.id.id_resulta);
29         id_salarioLiquido = findViewById(R.id.id_salarioLiquido);
30         id_descuen = findViewById(R.id.id_descuen);
31         id_ISSS = findViewById(R.id.id_ISSS);
32         id_AFP = findViewById(R.id.id_AFP);
33         id_RENTA = findViewById(R.id.id_RENTA);
34         id_gerente = findViewById(R.id.id_gerente);
35         id_asistente = findViewById(R.id.id_asistente);
36         id_secretaria = findViewById(R.id.id_secretaria);
37         id_otrosCargos = findViewById(R.id.id_otrosCargos);
38     }
39 }
```

```
Ejercicio_2.java x
37 id_otrosCargos = findViewById(R.id.id_otrosCargos);
38
39 }
40
41 public void Calcular (View in){
42     if (id_nombre.getText().toString().trim().isEmpty() || id_nombre.getText().toString().trim().isEmpty()){
43         Toast.makeText(context: this, text: "Ingresa los datos Solicitados", Toast.LENGTH_SHORT).show();
44     } else {
45         String nombres = id_nombre.getText().toString();
46         int horas = Integer.parseInt(id_horas.getText().toString());
47
48         double pago;
49         if (horas < 160){
50             pago = (horas) * 9.75;
51         } else {
52             pago= horas * 11.50;
53         }
54
55         double bonos, pagoIncremento, pagoFinal, descISSS, descAFP, descRENTA, totalDesc;
56         if (id_gerente.isChecked()){
57             bonos= 0.10;
58         } else if (id_asistente.isChecked()){
59             bonos = 0.05;
60         } else if (id_secretaria.isChecked()){
61             bonos = 0.03;
62         } else if (id_otrosCargos.isChecked()){
63             bonos = 0.02;
64         } else {
```

```
64         } else {
65             bonos = 0;
66         }
67         pagoIncremento= pago + pago * bonos;
68         descISSS = (pagoIncremento * 0.0525);
69         descAFP= (pagoIncremento*0.0688);
70         descRENTA = (pagoIncremento * 0.10);
71         totalDesc = descISSS + descAFP + descRENTA;
72         pagoFinal= pagoIncremento - totalDesc;
73
74
75         String id_resultado = "El sueldo base más bono de: " + nombres + " es:" + currency.format(pagoIncremento);
76         id_resulta.setText(id_resultado);
77
78         String id_iss = "El descuento del ISSS es: " + currency.format(descISSS);
79         id_ISSS.setText(id_iss);
80
81         String id_afp = "El descuento de la AFP es: " + currency.format(descAFP);
82         id_AFP.setText(id_afp);
83
84         String id_renta = "El descuento de la RENTA es: " + currency.format(descRENTA);
85         id_RENTA.setText(id_renta);
86
87         String id_liquido = "El salario liquido final menos los descuentos es de: " + currency.format(pagoFinal);
88         id_salarioLiquido.setText(id_liquido);
89
90     }
91 }
92 }
```

3:49

100%

Calculo de Salario de un Empleado

Alexander

180

Seleccione el Cargo del Empleado

☒ Gerente

☐ Asistente

☐ Secretaria

☐ Otros

CALCULAR

Resultado.

Listado de Descuentos

Salario Liquido Final

3:49

100%

Calculo de Salario de un Empleado

Alexander

180

Seleccione el Cargo del Empleado

☒ Gerente

☐ Asistente

☐ Secretaria

☐ Otros

CALCULAR

Resultado.

El sueldo base más bono de: Alexander es:\$2,277

Listado de Descuentos

El descuento del ISSS es: \$119.54

El descuento de la AFP es: \$156.66

El descuento de la RENTA es: \$227.7

Salario Liquido Final

El salario liquido final menos los descuentos es de: \$1,773.1



Horas trabajadas por el empleado son menores o igual de 160, deberán pagarse al empleado a \$9.75 cada hora.

3:50

100%

### Calculo de Salario de un Empleado

Alexander

150

Seleccione el Cargo del Empleado

☒

 Gerente

☐

 Asistente

☐

 Secretaria

☐

 Otros

CALCULAR

Resultado.

El sueldo base más bono de: Alexander es:\$1,608.75

Listado de Descuentos

El descuento del ISSS es: \$84.46

El descuento de la AFP es: \$110.68

El descuento de la RENTA es: \$160.88

Salario Liquido Final

El salario liquido final menos los descuentos es de: \$1,252.73