Prueba de Caja Blanca

“Seguimiento a graduados IASA-I”

Integrantes:

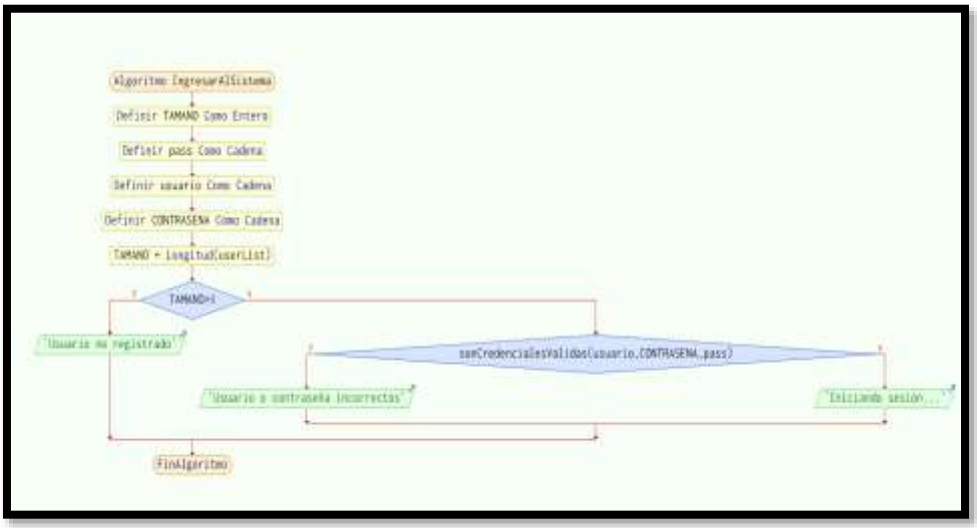
Genesis Calapaqui
Alex Paguay
Santiago Sañay

Fecha 2024-02-18

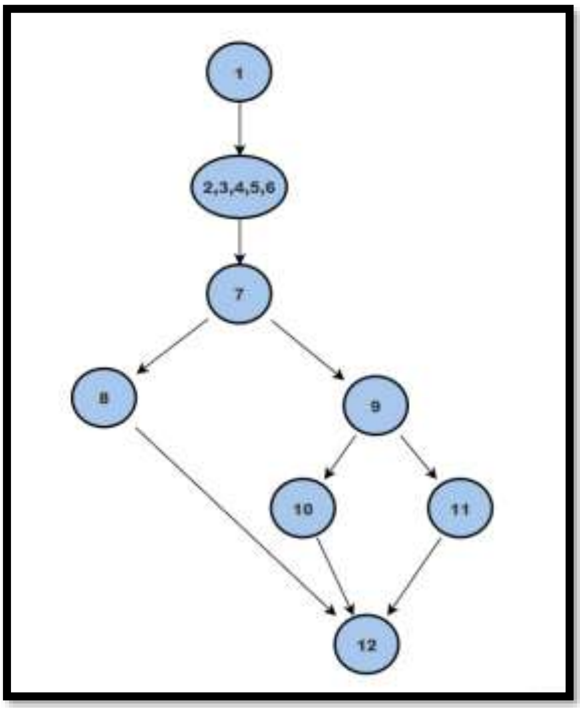
Req. 02: INGRESAR AL SISTEMA
CÓDIGO FUENTE

```
if (size>0){
    String pass=userList.get(0).getPassword();
    if (isValidCredentials(username, password,pass)) {
        Common.setUsername(userList.get(0).getUsername());
        Common.setPassword(userList.get(0).getPassword());
        Intent intent = new Intent(LoginActivity.this,
MainActivity.class);
        startActivity(intent);
        finish();
        errorTextView.setVisibility(View.GONE); // Ocultar el mensaje
de error si estaba visible
    } else {
        errorTextView.setVisibility(View.VISIBLE);
        errorTextView.setText("Usuario o contraseña incorrectos");
    }

}else{
    errorTextView.setVisibility(View.VISIBLE);
    errorTextView.setText("Usuario o contraseña incorrectos");
}
```

DIAGRAMA DE FLUJO


Algoritmo IngresarAlSistema
Definir TAMANO Como Entero
Definir pass Como Cadena
Definir usuario Como Cadena
Definir CONTRASENA Como Cadena
TAMANO ← Longitud(userList)
TAMANO>1
F
V
'Usuario no registrado'
sonCredencialesValidas(usuario,CONTRASENA,pass)
F
V
'Usuario o contraseña incorrectas'
'Iniciando sesión...'
FinAlgoritmo

GRAFO



RUTAS

R1: 1, 2, 3, 4,5,6,7,8,12
R2: 1, 2, 3, 4,5,6,7,9,10,12
R3: 1, 2, 3, 4,5,6,7,9,11,12

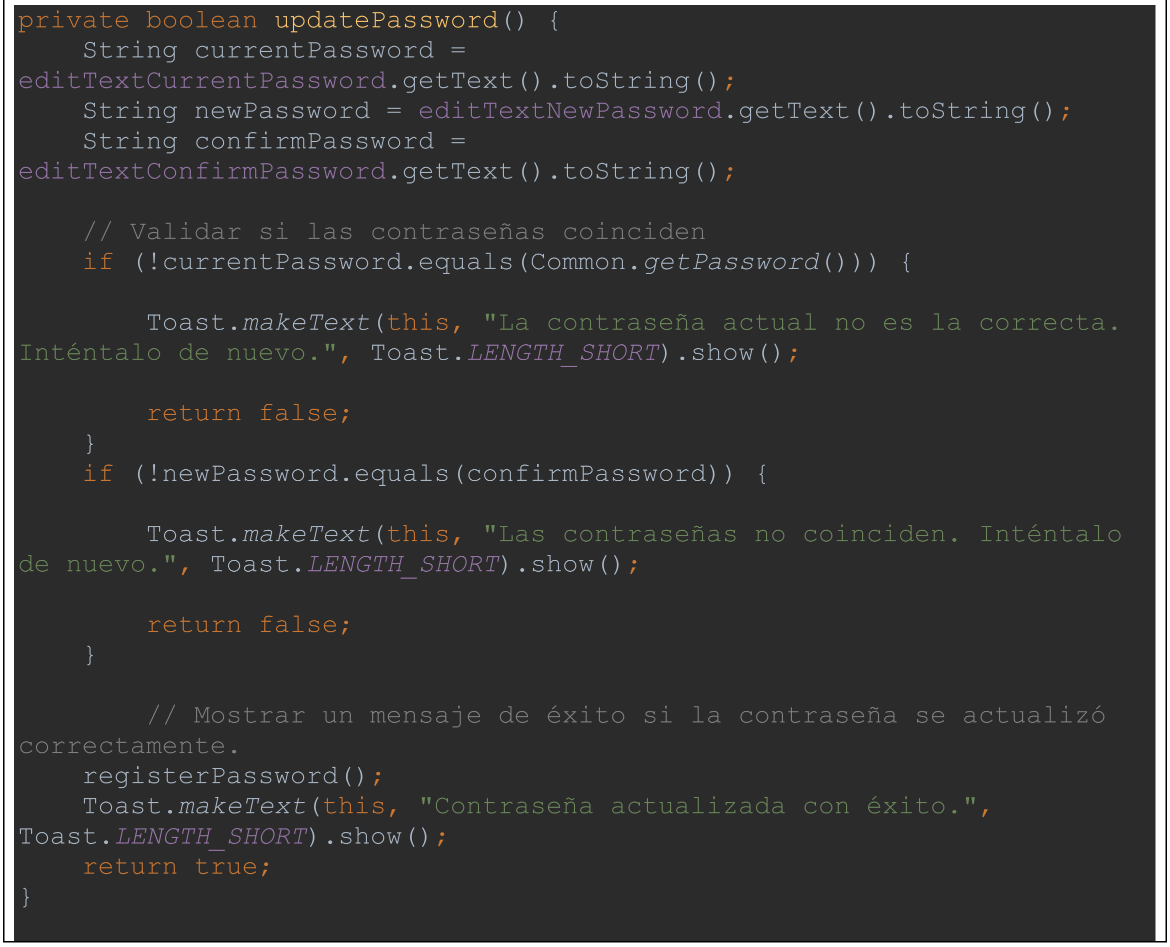
COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- V(G) = número de nodos predicados(decisiones)+1
  V(G)=2+1=3
- V(G) = A – N + 2
  V(G)= 9– 8 + 2 = 3

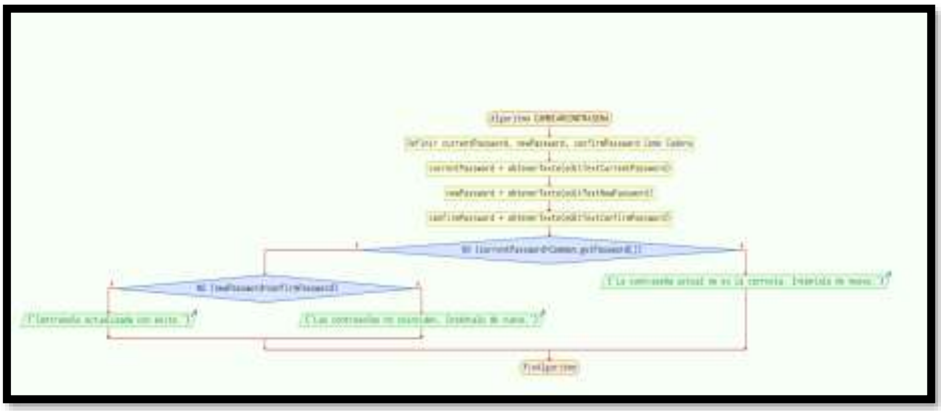
DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

Req. 03: CAMBIAR CONTRASEÑA
CÓDIGO FUENTE

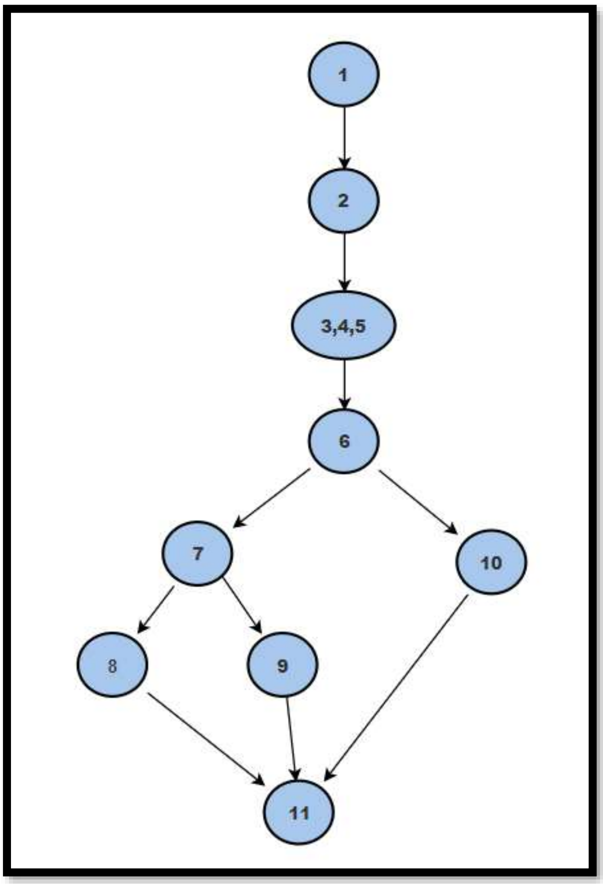


```
private boolean updatePassword() {
    String currentPassword =
editTextCurrentPassword.getText().toString();
    String newPassword = editTextNewPassword.getText().toString();
    String confirmPassword =
editTextConfirmPassword.getText().toString();

    // Validar si las contraseñas coinciden
    if (!currentPassword.equals(Common.getPassword())) {

        Toast.makeText(this, "La contraseña actual no es la correcta.
Inténtalo de nuevo.", Toast.LENGTH_SHORT).show();

        return false;
    }
    if (!newPassword.equals(confirmPassword)) {

        Toast.makeText(this, "Las contraseñas no coinciden. Inténtalo
de nuevo.", Toast.LENGTH_SHORT).show();

        return false;
    }

        // Mostrar un mensaje de éxito si la contraseña se actualizó
correctamente.
    registerPassword();
    Toast.makeText(this, "Contraseña actualizada con éxito.",
Toast.LENGTH_SHORT).show();
    return true;
}
```

DIAGRAMA DE FLUJO


Algoritmo CAMBIARCONTRASEÑA
Definir currentPassword, newPassword, confirmPassword Como Cadena
currentPassword ← obtenerTexto(editTextCurrentPassword)
newPassword ← obtenerTexto(editTextNewPassword)
confirmPassword ← obtenerTexto(editTextConfirmPassword)
Si (currentPassword=Common.getPassword())
Si (newPassword=confirmPassword)
('La contraseña actual no es la correcta. Inténtalo de nuevo.')
('Contraseña actualizada con éxito.')
('Las contraseñas no coinciden. Inténtalo de nuevo.')
FinAlgoritmo

GRAFO



RUTAS

R1: 1, 2, 3, 4,5,6,7,8,11
R2: 1, 2, 3, 4,5,6,7,9,11
R3: 1, 2, 3, 4,5,6,10,11

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

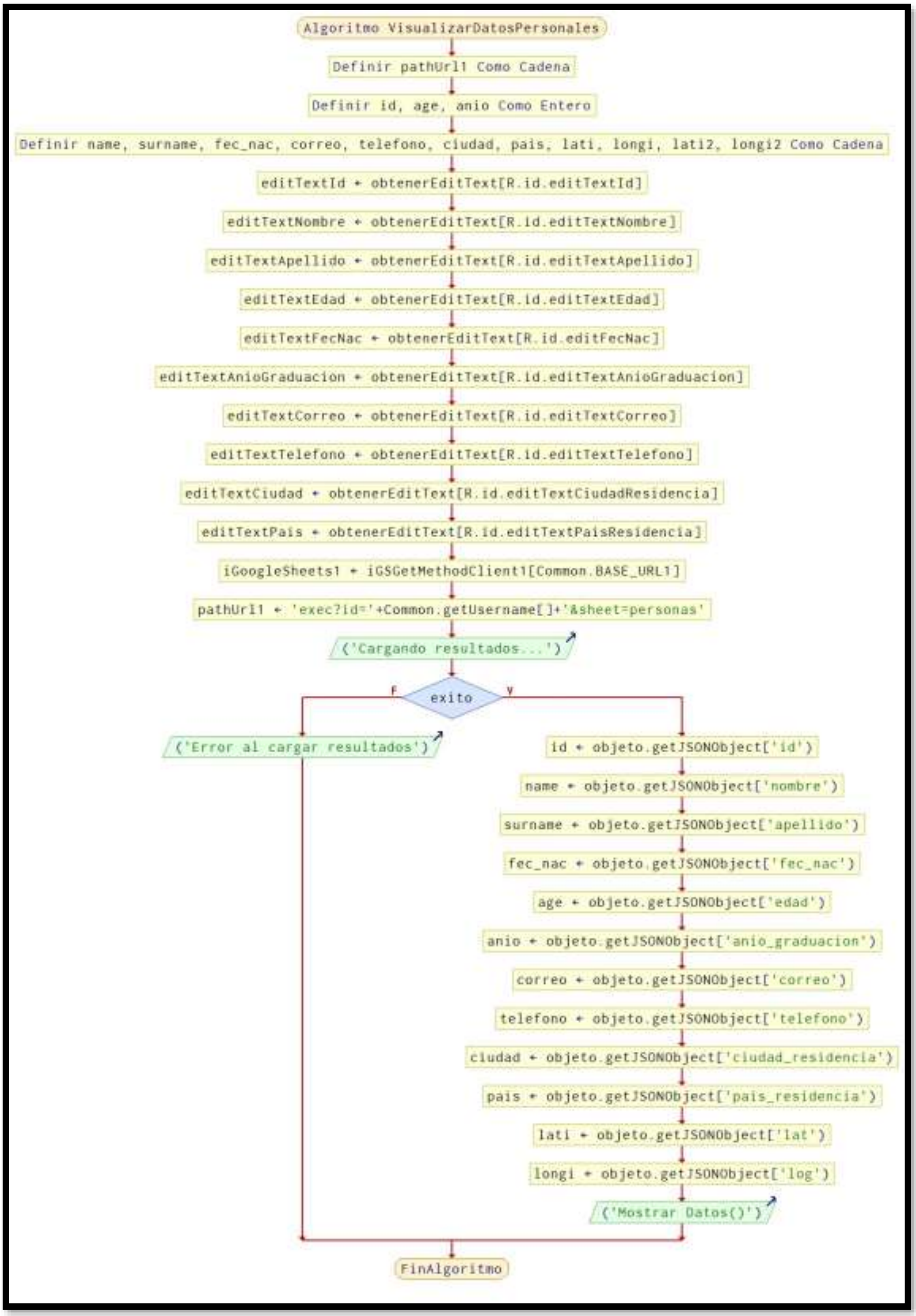
- V(G) = número de nodos predicados(decisiones)+1
  V(G)=2+1=3
- V(G) = A – N + 2
  V(G)= 10– 9 + 2 = 3

DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

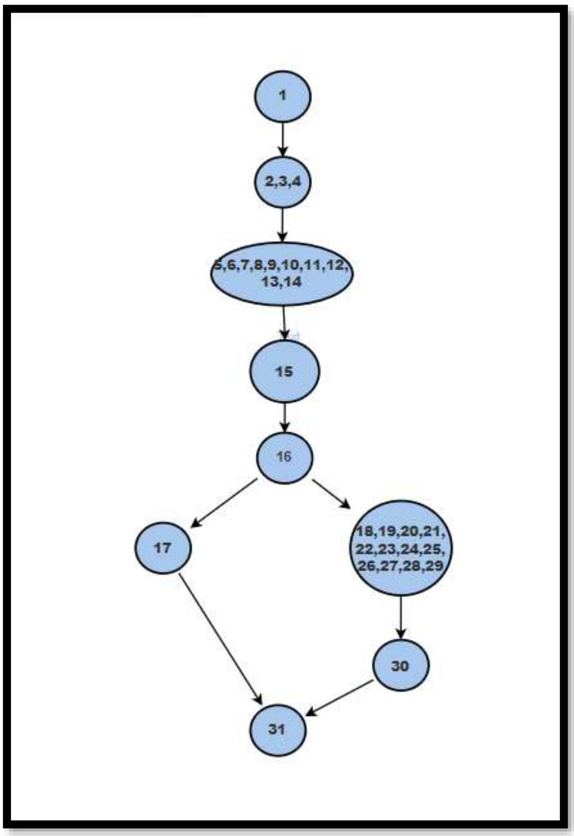
Req. 04: VISUALIZAR DATOS PERSONALES
CÓDIGO FUENTE

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_user_details);

    // Asigna los EditTexts a las variables de la clase
    editTextId = findViewById(R.id.editTextId);

    editTextNombre = findViewById(R.id.editTextNombre);
    editTextApellido = findViewById(R.id.editTextApellido);
    editTextEdad = findViewById(R.id.editTextEdad);
    editTextFecNac = findViewById(R.id.editFecNac);
    editTextAnioGraduacion=findViewById(R.id.editTextAnioGraduacion);
    editTextCorreo=findViewById(R.id.editTextCorreo);
    editTextTelefono=findViewById(R.id.editTextTelefono);
    editTextCiudad=findViewById(R.id.editTextCiudadResidencia);
    editTextPais=findViewById(R.id.editTextPaisResidencia);

    iGoogleSheets1 = Common.iGSGetMethodClient1(Common.BASE_URL1);
    String pathUrl1;
    pathUrl1 = "exec?id=" +
Common.getUsername().toString()+"&sheet=personas";

    String pathUrl;
    progressDialog = ProgressDialog.show(UserDetailsActivity.this,
            "Cargando resultados",
            "Espere por favor",
            true,
            false);
    try {
        //editTextId.setText(pathUrl1);

        iGoogleSheets1.getPeople(pathUrl1).enqueue(new
Callback<String>() {

            public void onResponse(@NonNull Call<String> call,
@NonNull Response<String> response) {
                try {

                    assert response.body() != null;
                    JSONObject responseObject = new
JSONObject(response.body());
                    JSONArray peopleArray =
responseObject.getJSONArray("persons");


//editTextId.setText(String.valueOf(peopleArray.length()));

                    JSONObject object = peopleArray.getJSONObject(0);

                    String id = object.getString("id");

                    String name = object.getString("nombre");

                    String surname = object.getString("apellido");
                    String fec_nac=object.getString("fec_nac");
                    String age = object.getString("edad");
                    String anio = object.getString("anio_graduacion");
                    String correo = object.getString("correo");
                    String telefono= object.getString("telefono");
                    String ciudad =
object.getString("ciudad_residencia");
                    String pais = object.getString("pais_residencia");

                    String lati=object.getString("lat");
                    String longi=object.getString("log");
                    String lati2=lati.replace(",",".");
                    String longi2=longi.replace(",",".");

                    double latitud = Double.parseDouble(lati2);
                    double longitud = Double.parseDouble(longi2);

                    Common.setLat(latitud);
                    Common.setLog(longitud);




                    editTextId.setText(id);
                    editTextNombre.setText(name);
                    editTextApellido.setText(surname);
                    editTextFecNac.setText(fec_nac);
                    editTextEdad.setText(age);
                    editTextAnioGraduacion.setText(anio);
                    editTextCorreo.setText(correo);
                    editTextTelefono.setText(telefono);
                    editTextCiudad.setText(ciudad);
                    editTextPais.setText(pais);

                    onMapReady(mMap);
                    progressDialog.dismiss();



                } catch (JSONException je) {
                    je.printStackTrace();
                }
            }

            @Override
            public void onFailure(@NonNull Call<String> call, @NonNull
Throwable t) {

            }
        });
    } catch (Exception e) {
        e.printStackTrace();
    }


    // Asigna el Listener al botón de regreso
    buttonBack = findViewById(R.id.buttonBack);
    buttonBack.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // Regresar al MainActivity
            finish();
        }
    });
}
```

DIAGRAMA DE FLUJO


Algoritmo VisualizarDatosPersonales
Definir pathUrl1 Como Cadena
Definir id, age, anio Como Entero
Definir name, surname, fec_nac, correo, telefono, ciudad, pais, lati, longi, lati2, longi2 Como Cadena
editTextId ← obtenerEditText[R.id.editTextId]
editTextNombre ← obtenerEditText[R.id.editTextNombre]
editTextApellido ← obtenerEditText[R.id.editTextApellido]
editTextEdad ← obtenerEditText[R.id.editTextEdad]
editTextFecNac ← obtenerEditText[R.id.editFecNac]
editTextAnioGraduacion ← obtenerEditText[R.id.editTextAnioGraduacion]
editTextCorreo ← obtenerEditText[R.id.editTextCorreo]
editTextTelefono ← obtenerEditText[R.id.editTextTelefono]
editTextCiudad ← obtenerEditText[R.id.editTextCiudadResidencia]
editTextPais ← obtenerEditText[R.id.editTextPaisResidencia]
iGoogleSheets1 ← iGSGetMethodClient1[Common.BASE_URL1]
pathUrl1 ← 'exec?id='+Common.getUsername[]+'&sheet=personas'
('Cargando resultados...')
exito
F
V
('Error al cargar resultados')
id ← objeto.getJSONObject['id')
name ← objeto.getJSONObject['nombre')
surname ← objeto.getJSONObject['apellido')
fec_nac ← objeto.getJSONObject['fec_nac')
age ← objeto.getJSONObject['edad')
anio ← objeto.getJSONObject['anio_graduacion')
correo ← objeto.getJSONObject['correo')
telefono ← objeto.getJSONObject['telefono')
ciudad ← objeto.getJSONObject['ciudad_residencia')
pais ← objeto.getJSONObject['pais_residencia')
lati ← objeto.getJSONObject['lat')
longi ← objeto.getJSONObject['log')
('Mostrar Datos()')
FinAlgoritmo

GRAFO



RUTAS

R1: 1, 2, 3, 4,5,6,7,8,9,10,11,12,13,14,15,16,17,31
R2: 1, 2, 3, 4,5,6,7,8,9,10,11,12,13,14,15,16,18,19,20,21,22,23,24,25,26,27,28,29,30,31

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

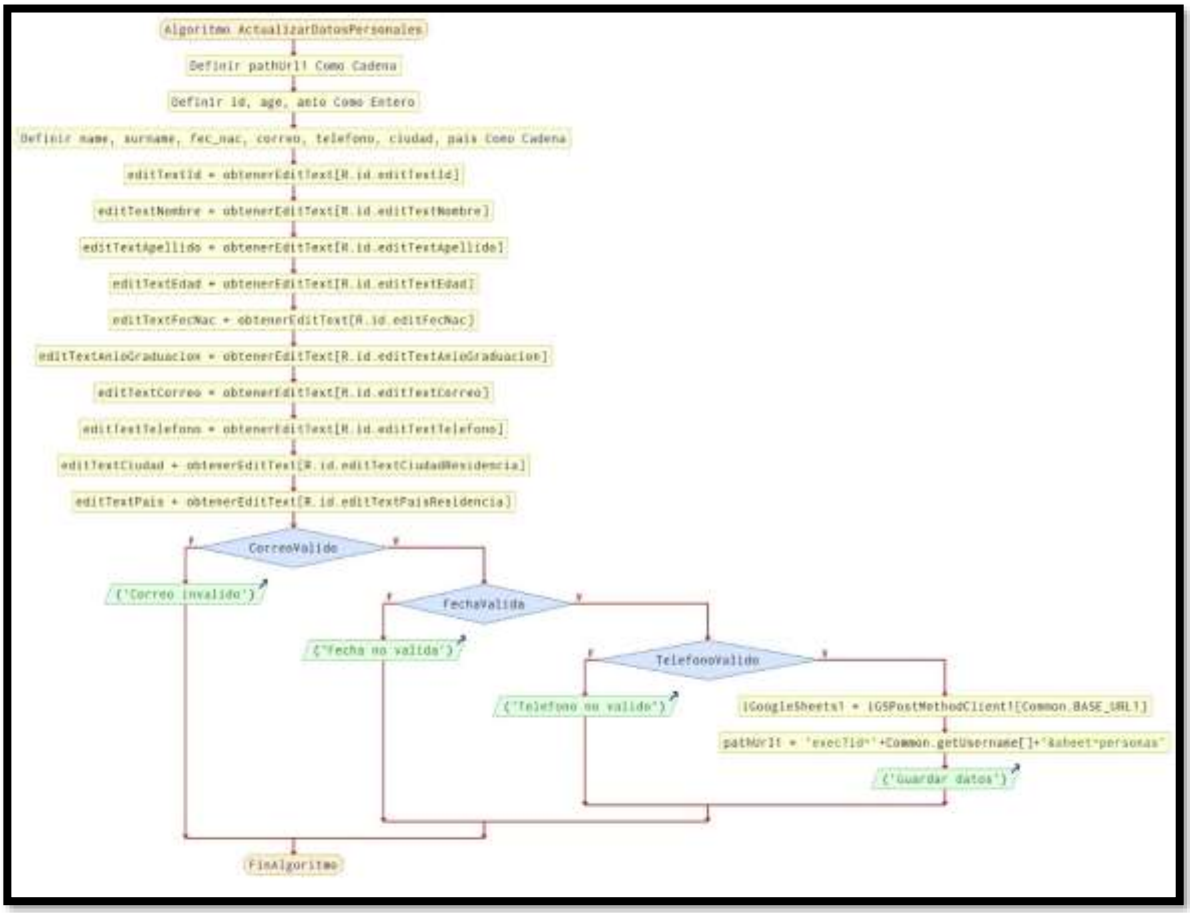
- V(G) = número de nodos predicados(decisiones)+1
  V(G)=1+1=2
- V(G) = A – N + 2
  V(G)= 9 – 9 + 2 = 2

DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

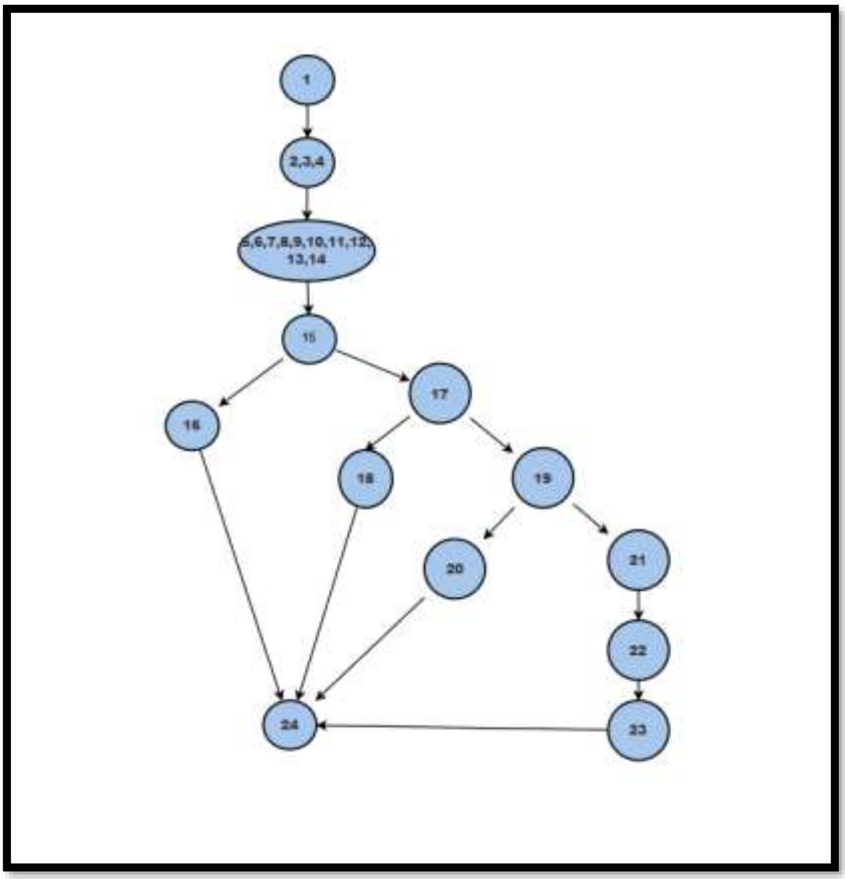
Req. 05: ACTUALIZAR DATOS PERSONALES
CÓDIGO FUENTE

```
    public void afterTextChanged(Editable editable) {
        // Validar si el texto ingresado es una dirección de correo 
electrónico válida
        boolean isEmailValid = isValidEmail(editable.toString());

        if (isEmailValid) {
            // El correo electrónico es válido, limpiamos el error si 
estaba presente
            editTextCorreo.setError(null);
        } else {
            // El correo electrónico no es válido, mostramos un 
mensaje de error
            editTextCorreo.setError("Correo electrónico inválido");
        }
        // Habilitar o deshabilitar el botón según la validez del 
correo y el teléfono
        enableSaveButton(isEmailValid, 
isValidPhone(editTextTelefono.getText().toString()));
    }
});

// Agregar un TextWatcher al EditText de teléfono para validar el 
teléfono en tiempo real
editTextTelefono.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence charSequence, int i, 
int i1, int i2) {
    }

    @Override
    public void onTextChanged(CharSequence charSequence, int i, int 
i1, int i2) {
    }

    @Override
    public void afterTextChanged(Editable editable) {
        // Validar si el texto ingresado es un teléfono válido
        boolean isPhoneValid = isValidPhone(editable.toString());

        if (isPhoneValid) {
            // El teléfono es válido, limpiamos el error si estaba 
presente
            editTextTelefono.setError(null);
        } else {
            // El teléfono no es válido, mostramos un mensaje de error
            editTextTelefono.setError("Teléfono inválido");
        }
        // Habilitar o deshabilitar el botón según la validez del 
correo y el teléfono

enableSaveButton(isValidEmail(editTextCorreo.getText().toString()), 
isPhoneValid);
    }
});
```

DIAGRAMA DE FLUJO


Algoritmo ActualizarDatosPersonales
Definir pathUrl1 Como Cadena
Definir id, age, anio Como Entero
Definir name, surname, fec_nac, correo, telefono, ciudad, pais Como Cadena
editTextId = obtenerEditText[R.id.editTextId]
editTextNombre = obtenerEditText[R.id.editTextNombre]
editTextApellido = obtenerEditText[R.id.editTextApellido]
editTextEdad = obtenerEditText[R.id.editTextEdad]
editTextFecNac = obtenerEditText[R.id.editFecNac]
editTextAnioGraduacion = obtenerEditText[R.id.editTextAnioGraduacion]
editTextCorreo = obtenerEditText[R.id.editTextCorreo]
editTextTelefono = obtenerEditText[R.id.editTextTelefono]
editTextCiudad = obtenerEditText[R.id.editTextCiudadResidencia]
editTextPais = obtenerEditText[R.id.editTextPaisResidencia]
f
CorreoValido
v
{'Correo invalido'}
f
FechaValida
v
{'Fecha no valida'}
f
TelefonoValido
v
{'Telefono no valido'}
iGoogleSheets1 = iGSPostMethodClient1[Common.BASE_URL1]
pathUrl1 = 'exec?id='+Common.getUsername[]+'&sheet=personas'
{'Guardar datos'}
FinAlgoritmo

GRAFO



RUTAS

R1: 1, 2, 3, 4,5,6,7,8,9,10,11,12,13,14,15,16,24
R2: 1, 2, 3, 4,5,6,7,8,9,10,11,12,13,14,15,17,18,24
R3: 1, 2, 3, 4,5,6,7,8,9,10,11,12,13,14,15,17,19,20,24
R4: 1, 2, 3, 4,5,6,7,8,9,10,11,12,13,14,15,17,19,21,22,23,24

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

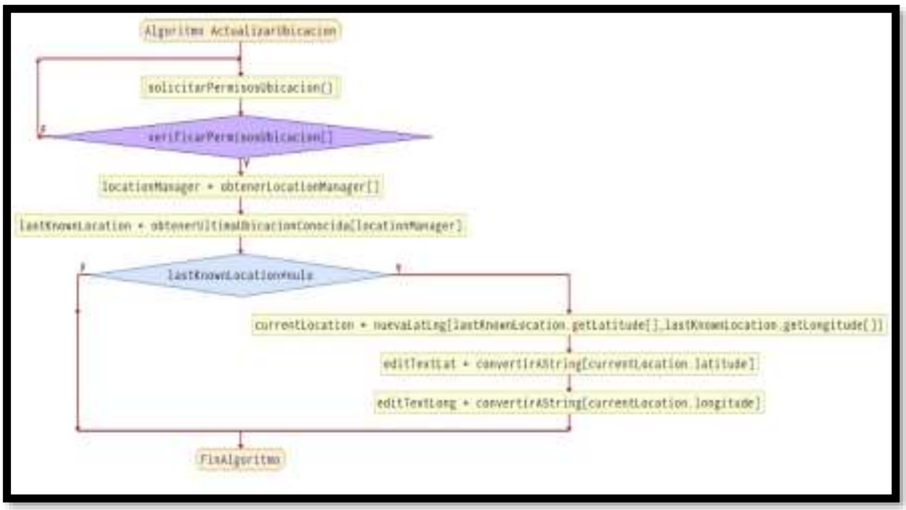
- V(G) = número de nodos predicados(decisiones)+1
  V(G)=3+1=4
- V(G) = A – N + 2
  V(G)= 15 – 13 + 2 = 4

DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

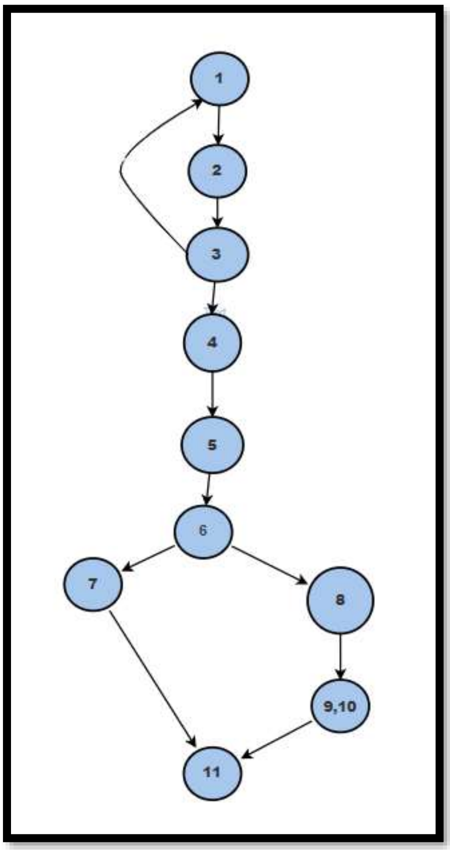
Req. 06: ACTUALIZAR UBICACIÓN GEOGRÁFICA
CÓDIGO FUENTE

```
    private void centerMapOnMyLocation() {

    do{
// Solicitar permisos de ubicación si no están otorgados
        ActivityCompat.requestPermissions(this,
                new
String[]{Manifest.permission.ACCESS_FINE_LOCATION},

    }while(ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION)
            != PackageManager.PERMISSION_GRANTED);
    // Verificar permisos de ubicación

        // Obtener la ubicación actual
        LocationManager locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
        Location lastKnownLocation =
locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        if (lastKnownLocation != null) {
            LatLng currentLocation = new
LatLng(lastKnownLocation.getLatitude(),
lastKnownLocation.getLongitude());

mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(currentLocation,
15));

editTextLat.setText(String.valueOf(currentLocation.latitude));

editTextLong.setText(String.valueOf(currentLocation.longitude));
        }

}
```

DIAGRAMA DE FLUJO


Algoritmo ActualizarUbicacion
solicitarPermisosUbicacion()
F
verificarPermisosUbicacion()
V
locationManager ← obtenerLocationManager()
lastKnownLocation ← obtenerUltimaUbicacionConocida(locationManager)
F
lastKnownLocation≠nulo
V
currentLocation ← nuevaLatLng(lastKnownLocation.getLatitude(),lastKnownLocation.getLongitude())
editTextLat ← convertirAString(currentLocation.latitude)
editTextLong ← convertirAString(currentLocation.longitude)
FinAlgoritmo

GRAFO



RUTAS

R1: 1, 2, 3, 1
R2: 1, 2, 3, 4,5,6,7,11
R3: 1, 2, 3, 4,5,6,8,9,10,11

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- V(G) = número de nodos predicados(decisiones)+1
  V(G)=2+1=3
- V(G) = A – N + 2
  V(G)= 11 – 10+ 2 = 3

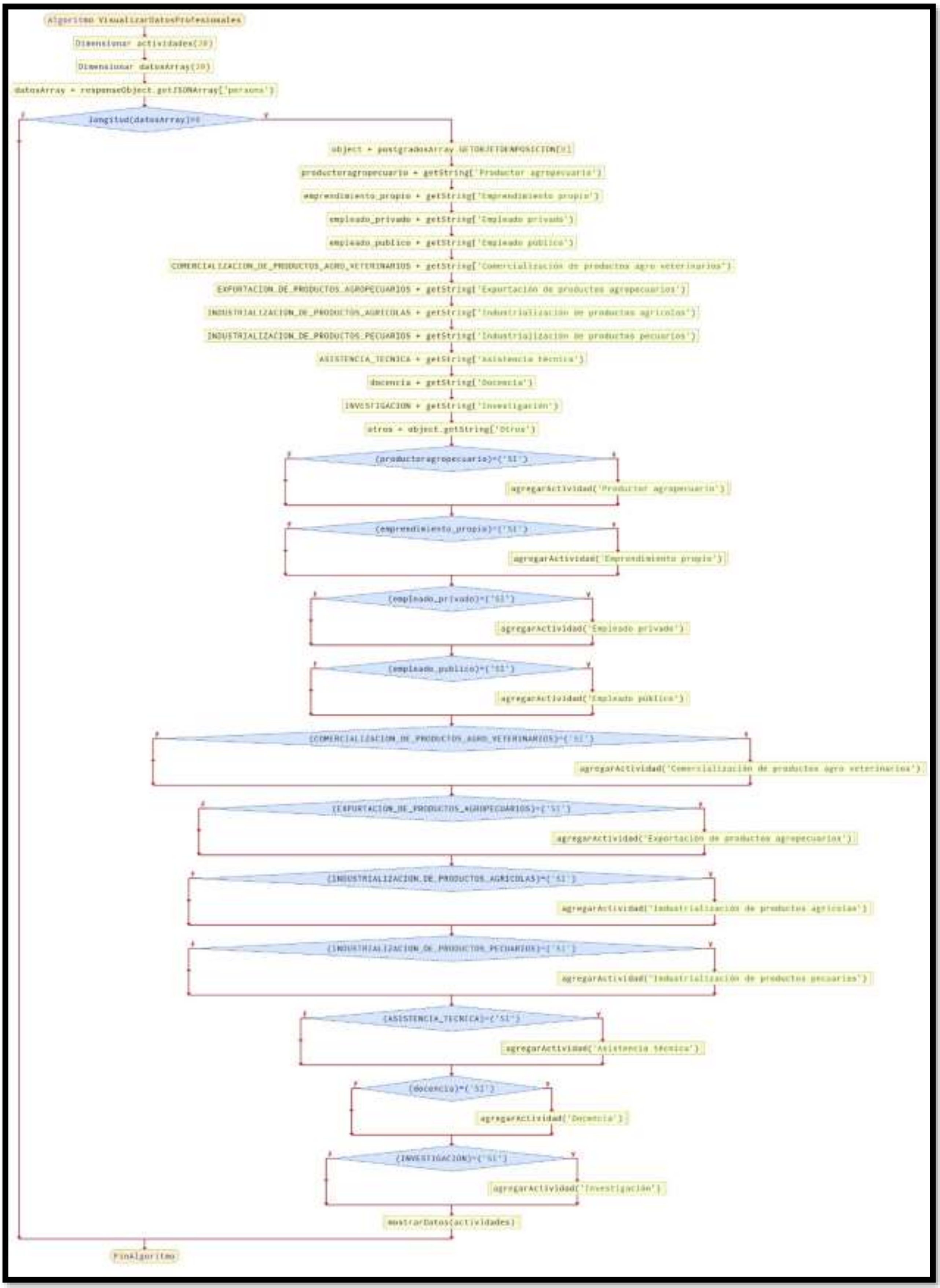
DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

Req. 07: VISUALIZAR INFORMACIÓN PROFESIONAL
CÓDIGO FUENTE

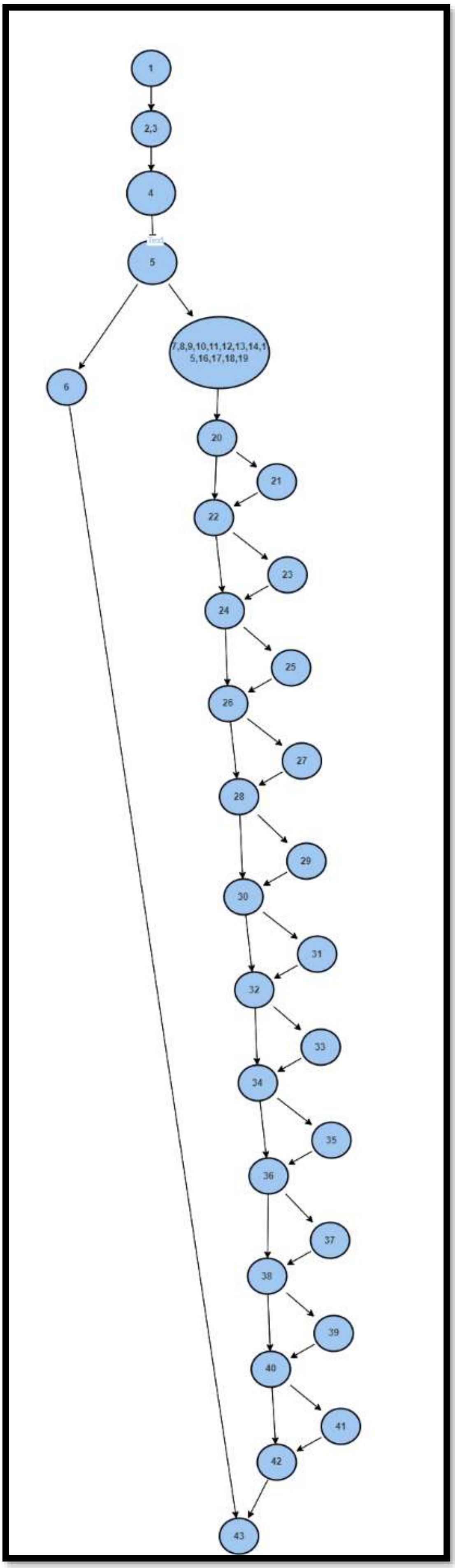
```
JSONArray postgradosArray = responseObject.getJSONArray("persons");
if(postgradosArray.length()>0){
    JSONObject object = postgradosArray.getJSONObject(0);
    String productoragropecuario = object.getString("Productor
agropecuario");
    String emprendimiento_propio = object.getString("Emprendimiento
propio");
    String empleado_privado = object.getString("Empleado privado");
    String empleado_publico = object.getString("Empleado público");
    String comercialización_de_productos_agro_veterinarios =
object.getString("Comercialización de productos agro veterinarios");
    String exportación_de_productos_agropecuarios =
object.getString("Exportación de productos agropecuarios");
    String industrialización_de_productos_agrícolas =
object.getString("Industrialización de productos agrícolas");
    String industrialización_de_productos_pecuarios =
object.getString("Industrialización de productos pecuarios");
    String asistencia_técnica = object.getString("Asistencia
técnica");
    String docencia = object.getString("Docencia");
    String investigación = object.getString("Investigación");
    String otros = object.getString("Otros");

    if(productoragropecuario.equals("SI")){
        actividades.add("Productor agropecuario");
    }
    if(emprendimiento_propio.equals("SI")){
        actividades.add("Emprendimiento propio");
    }
    if(empleado_privado.equals("SI")){
        actividades.add("Empleado privado");
    }
    if(empleado_publico.equals("SI")){
        actividades.add("Empleado público");
    }
    if(comercialización_de_productos_agro_veterinarios.equals("SI")){
        actividades.add("Comercialización de productos agro
veterinarios");
    }
    if(exportación_de_productos_agropecuarios.equals("SI")){
        actividades.add("Exportación de productos agropecuarios");
    }
    if(industrialización_de_productos_agrícolas.equals("SI")){
        actividades.add("Industrialización de productos agrícolas");
    }
    if(industrialización_de_productos_pecuarios.equals("SI")){
        actividades.add("Industrialización de productos pecuarios");
    }
    if(asistencia_técnica.equals("SI")){
        actividades.add("Asistencia técnica");
    }
    if(docencia.equals("SI")){
        actividades.add("Docencia");
    }
    if(investigación.equals("SI")){
        actividades.add("Investigación");
    }

}
setProfesionalAdapter(actividades);
```

[redacted]
[redacted]

DIAGRAMA DE FLUJO


Algoritmo VisualizarDatosProfesionales
Dimensionar actividades(20)
Dimensionar datosArray(20)
datosArray = responseObject.getJSONArray['persona']
longitud(datosArray)>0
F
V
object ← postgradosArray.GETOBJETOENPOSICION[0]
productoragropecuario ← getString['Productor agropecuario']
emprendimiento_propio ← getString['Emprendimiento propio']
empleado_privado ← getString['Empleado privado']
empleado_publico ← getString['Empleado público']
COMERCIALIZACION_DE_PRODUCTOS_AGRO_VETERINARIOS ← getString['Comercialización de productos agro veterinarios']
EXPORTACION_DE_PRODUCTOS_AGROPECUARIOS ← getString['Exportación de productos agropecuarios']
INDUSTRIALIZACION_DE_PRODUCTOS_AGRICOLAS ← getString['Industrialización de productos agrícolas']
INDUSTRIALIZACION_DE_PRODUCTOS_PECUARIOS ← getString['Industrialización de productos pecuarios']
ASISTENCIA_TECNICA ← getString['Asistencia técnica']
docencia ← getString['Docencia']
INVESTIGACION ← getString['Investigación']
otros ← object.getString['Otros']
(productoragropecuario)=('SI')
agregarActividad('Productor agropecuario')
(emprendimiento_propio)=('SI')
agregarActividad('Emprendimiento propio')
(empleado_privado)=('SI')
agregarActividad('Empleado privado')
(empleado_publico)=('SI')
agregarActividad('Empleado público')
(COMERCIALIZACION_DE_PRODUCTOS_AGRO_VETERINARIOS)=('SI')
agregarActividad('Comercialización de productos agro veterinarios')
(EXPORTACION_DE_PRODUCTOS_AGROPECUARIOS)=('SI')
agregarActividad('Exportación de productos agropecuarios')
(INDUSTRIALIZACION_DE_PRODUCTOS_AGRICOLAS)=('SI')
agregarActividad('Industrialización de productos agrícolas')
(INDUSTRIALIZACION_DE_PRODUCTOS_PECUARIOS)=('SI')
agregarActividad('Industrialización de productos pecuarios')
(ASISTENCIA_TECNICA)=('SI')
agregarActividad('Asistencia técnica')
(docencia)=('SI')
agregarActividad('Docencia')
(INVESTIGACION)=('SI')
agregarActividad('Investigación')
mostrarDatos(actividades)
FinAlgoritmo

GRAFO


1
2,3
4
Text
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
32
33
34
35
36
37
38
39
40
41
42
43

RUTAS

R1: 1,2,3,4,5,6,43
R2:1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,
34,35,36,37,38,39,40,41,42,43
R3:1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,24,26,28,30,32,34,36,38,40,42,43
R4:1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,22,23,24,26,28,30,32,34,36,38,40,42,43
R5:1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,22,24,25,26,28,30,32,34,36,38,40,42,43
R6:1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,22,24,25,26,28,30,32,34,36,38,40,42,43
R7:1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,22,24,26,27,28,30,32,34,36,38,40,42,43
R8:1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,22,24,26,28,29,30,32,34,36,38,40,42,43
R9:1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,22,24,26,28,30,31,32,34,36,38,40,42,43
R10:1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,22,24,26,28,30,32,33,34,36,38,40,42,43
R11:1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,22,24,26,28,30,32,34,35,36,38,40,42,43
R12:1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,22,24,26,28,30,32,34,36,37,38,40,42,43
R13:1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,22,24,26,28,30,32,34,36,38,39,40,42,43

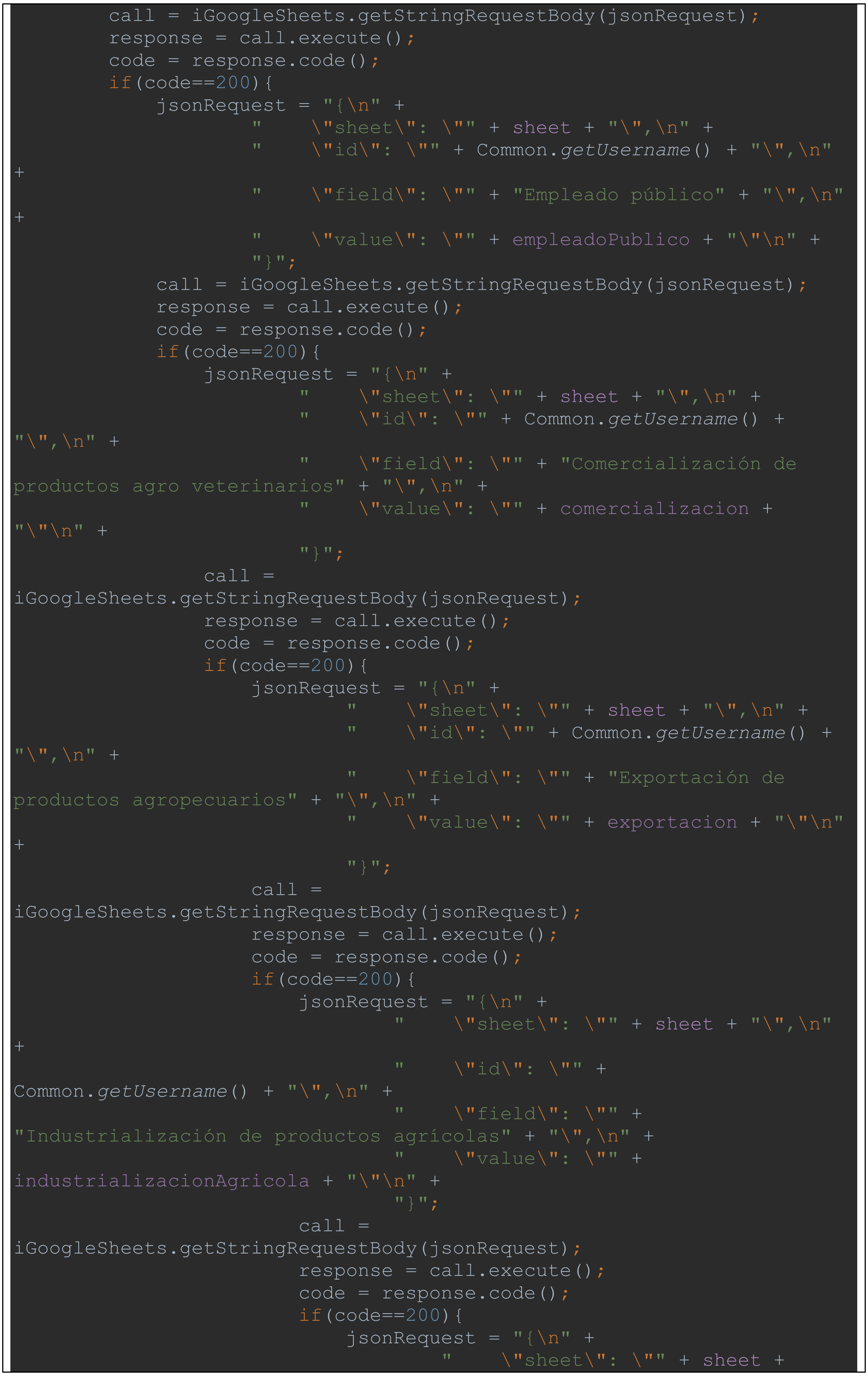
COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- V(G) = número de nodos predicados(decisiones)+1
  V(G)=12+1=13
- V(G) = A – N + 2
  V(G)= 41 – 30+ 2 = 13

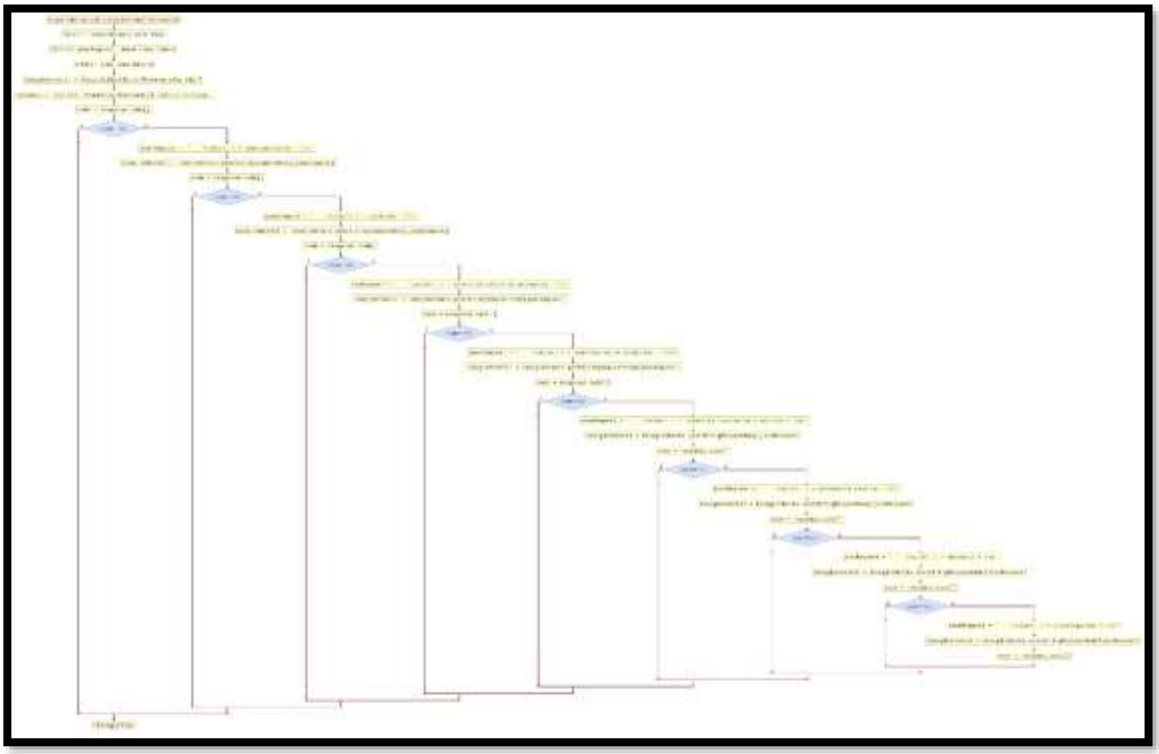
DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

Req. 08: ACTUALIZAR CAMPO PROFESIONAL
CÓDIGO FUENTE

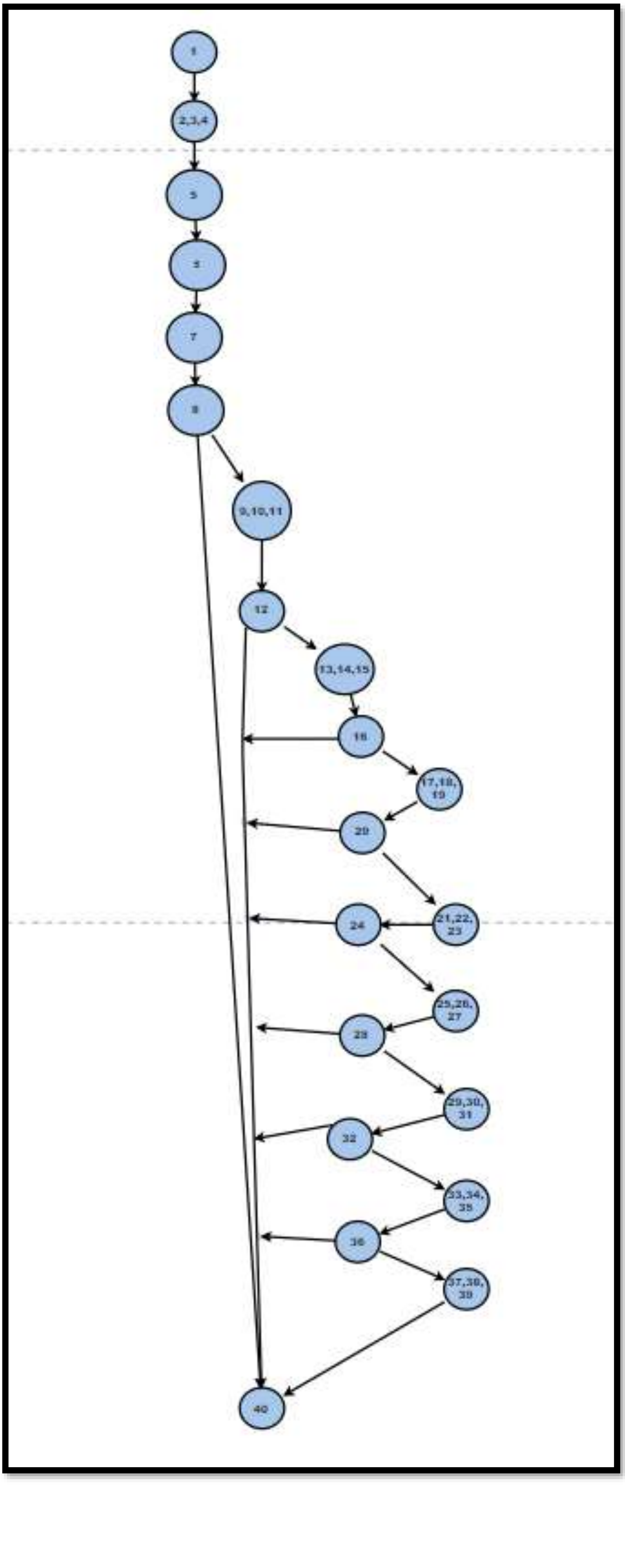


```
        call = iGoogleSheets.getStringRequestBody(jsonRequest);
        response = call.execute();
        code = response.code();
        if(code==200){
            jsonRequest = "{\n" +
                    "    \"sheet\": \"" + sheet + "\",\n" +
                    "    \"id\": \"" + Common.getUsername() + "\",\n"
+
                    "    \"field\": \"" + "Empleado público" + "\",\n"
+
                    "    \"value\": \"" + empleadoPublico + "\"\n" +
                    "}";
            call = iGoogleSheets.getStringRequestBody(jsonRequest);
            response = call.execute();
            code = response.code();
            if(code==200){
                jsonRequest = "{\n" +
                        "    \"sheet\": \"" + sheet + "\",\n" +
                        "    \"id\": \"" + Common.getUsername() +
"\",\n" +
                        "    \"field\": \"" + "Comercialización de
productos agro veterinarios" + "\",\n" +
                        "    \"value\": \"" + comercializacion +
"\"\n" +
                        "}";
                call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                response = call.execute();
                code = response.code();
                if(code==200){
                    jsonRequest = "{\n" +
                            "    \"sheet\": \"" + sheet + "\",\n" +
                            "    \"id\": \"" + Common.getUsername() +
"\",\n" +
                            "    \"field\": \"" + "Exportación de
productos agropecuarios" + "\",\n" +
                            "    \"value\": \"" + exportacion + "\"\n"
+
                            "}";
                    call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                    response = call.execute();
                    code = response.code();
                    if(code==200){
                        jsonRequest = "{\n" +
                                "    \"sheet\": \"" + sheet + "\",\n"
+
                                "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                "    \"field\": \"" +
"Industrialización de productos agrícolas" + "\",\n" +
                                "    \"value\": \"" +
industrializacionAgricola + "\"\n" +
                                "}";
                        call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                        response = call.execute();
                        code = response.code();
                        if(code==200){
                            jsonRequest = "{\n" +
                                    "    \"sheet\": \"" + sheet +
"\",\n" +
                                    "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                    "    \"field\": \"" +
"Industrialización de productos pecuarios" + "\",\n" +
                                    "    \"value\": \"" +
industrializacionPecuarios + "\"\n" +
                                    "}";
                            call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                            response = call.execute();
                            code = response.code();
                            if(code==200){
                                jsonRequest = "{\n" +
                                        "    \"sheet\": \"" + sheet +
"\",\n" +
                                        "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                        "    \"field\": \"" +
"Asistencia técnica" + "\",\n" +
                                        "    \"value\": \"" +
asistenciaTecnica + "\"\n" +
                                        "}";
                                call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                response = call.execute();
                                code = response.code();
                                if(code==200){
                                    jsonRequest = "{\n" +
                                            "    \"sheet\": \"" +
sheet + "\",\n" +
                                            "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                            "    \"field\": \"" +
"Docencia" + "\",\n" +
                                            "    \"value\": \"" +
docencia + "\"\n" +
                                            "}";
                                    call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                    response = call.execute();
                                    code = response.code();
                                    if(code==200){
                                        jsonRequest = "{\n" +
                                                "    \"sheet\": \"" +
sheet + "\",\n" +
                                                "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                                "    \"field\": \"" +
"Investigación" + "\",\n" +
                                                "    \"value\": \"" +
investigacion + "\"\n" +
                                                "}";
                                        call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                        response = call.execute();
                                        code = response.code();
```

DIAGRAMA DE FLUJO



GRAFO


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
32
33,34,
35
36
37,38,
39
40

RUTAS

R1: 1, 2, 3, 4,5,6,7,8,40
R2: 1,2,3,4,5,6,7,8,9,10,11,12,40
R3:1,2,3,4,5,6,7,8,9,10,11,12,40
R4:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,40
R5:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,40
R6:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,40
R7:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,40
R7:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,40
R8:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,40
R9:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40

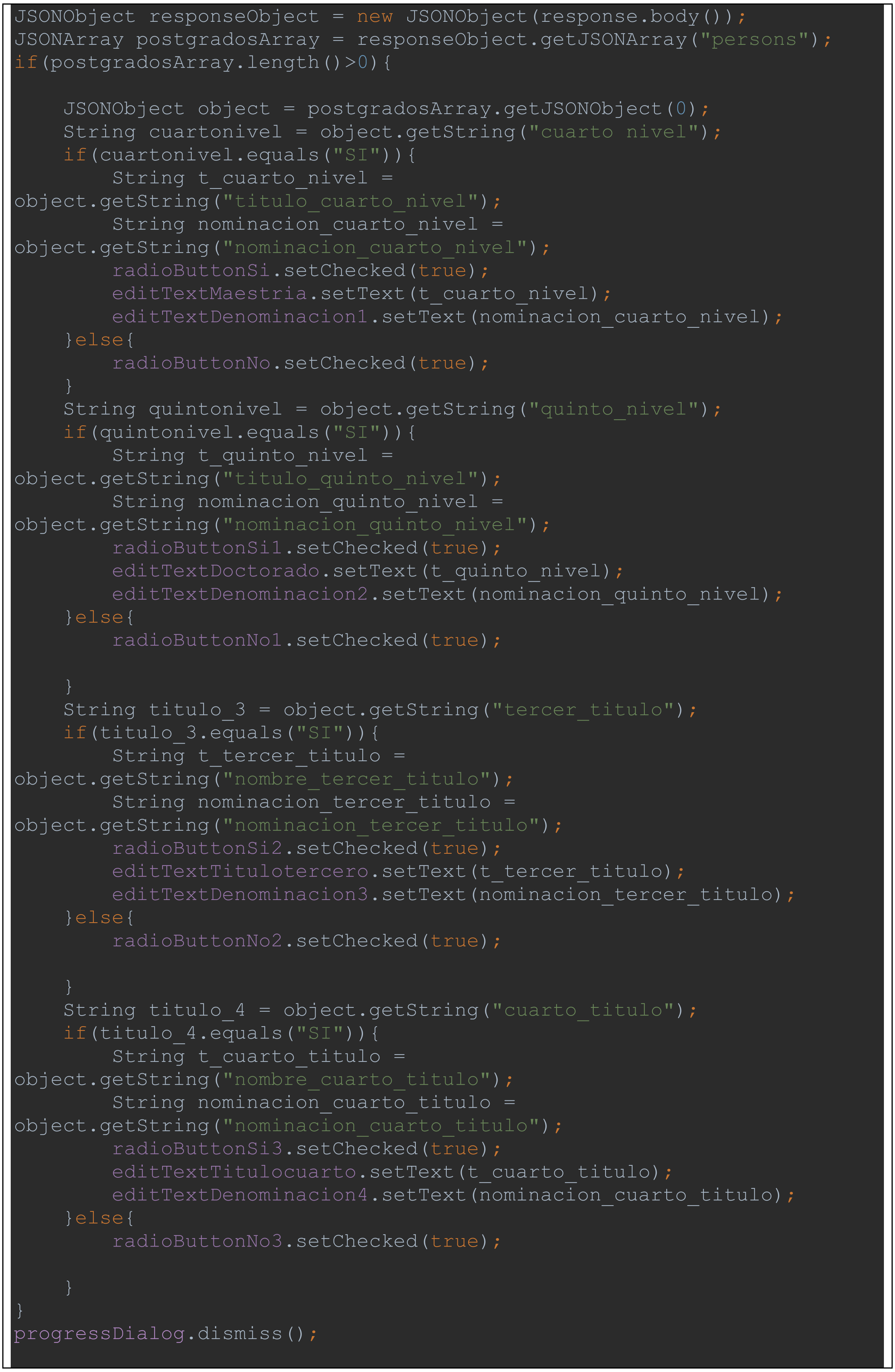
COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- V(G) = número de nodos predicados(decisiones)+1
  V(G)=8+1=9
- V(G) = A – N + 2
  V(G)= 29 – 22+ 2 = 9

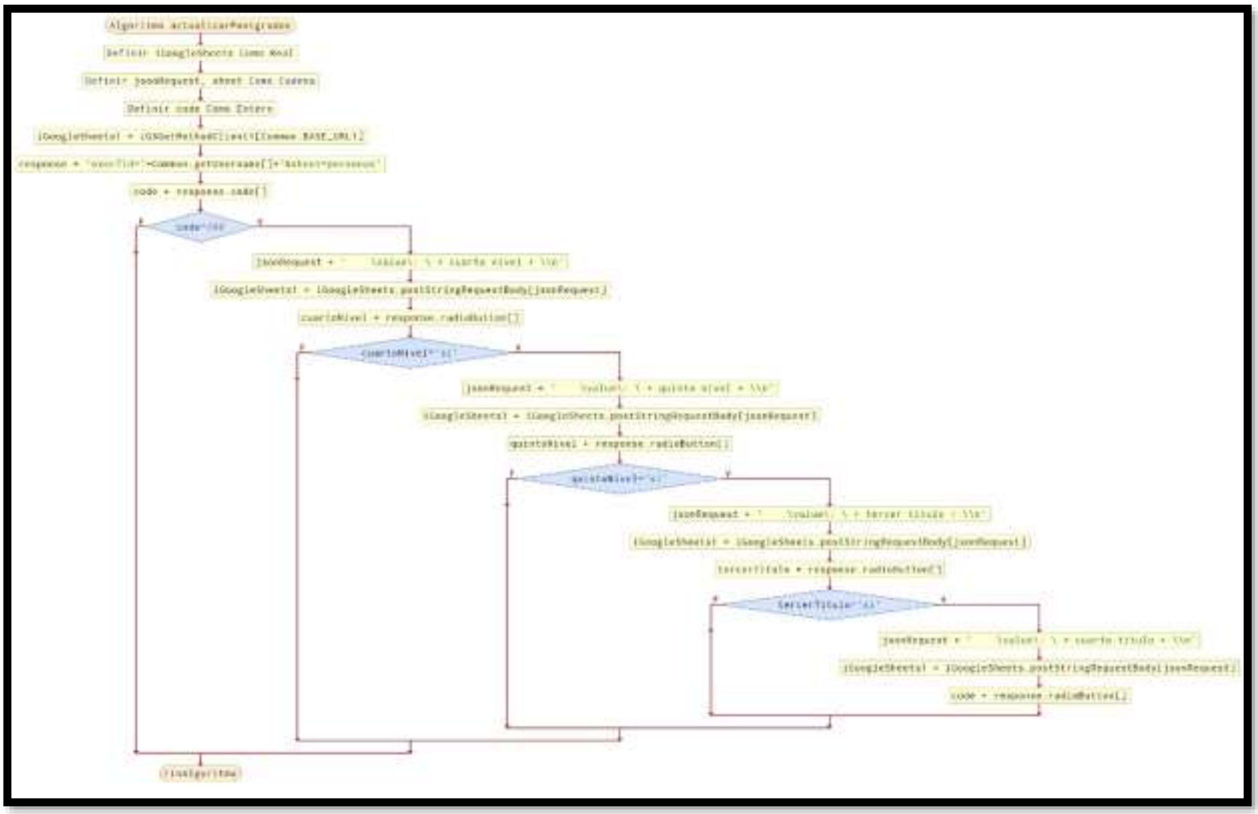
DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

Req. 09: ACTUALIZAR INFORMACIÓN DE POSTGRADOS
CÓDIGO FUENTE

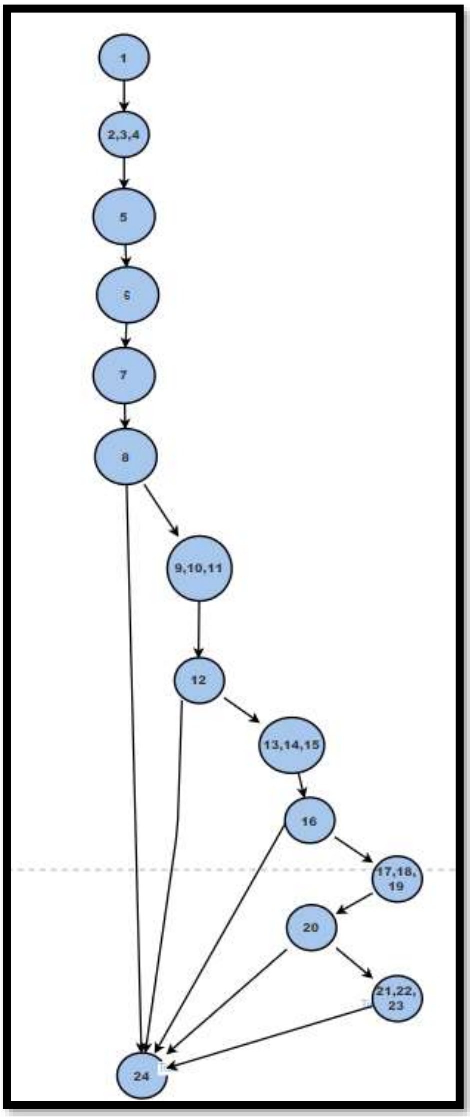


```
JSONObject responseObject = new JSONObject(response.body());
JSONArray postgradosArray = responseObject.getJSONArray("persons");
if(postgradosArray.length()>0){

    JSONObject object = postgradosArray.getJSONObject(0);
    String cuartonivel = object.getString("cuarto nivel");
    if(cuartonivel.equals("SI")){
        String t_cuarto_nivel =
object.getString("titulo_cuarto_nivel");
        String nominacion_cuarto_nivel =
object.getString("nominacion_cuarto_nivel");
        radioButtonSi.setChecked(true);
        editTextMaestria.setText(t_cuarto_nivel);
        editTextDenominacion1.setText(nominacion_cuarto_nivel);
    }else{
        radioButtonNo.setChecked(true);
    }
    String quintonivel = object.getString("quinto_nivel");
    if(quintonivel.equals("SI")){
        String t_quinto_nivel =
object.getString("titulo_quinto_nivel");
        String nominacion_quinto_nivel =
object.getString("nominacion_quinto_nivel");
        radioButtonSi1.setChecked(true);
        editTextDoctorado.setText(t_quinto_nivel);
        editTextDenominacion2.setText(nominacion_quinto_nivel);
    }else{
        radioButtonNo1.setChecked(true);

    }
    String titulo_3 = object.getString("tercer_titulo");
    if(titulo_3.equals("SI")){
        String t_tercer_titulo =
object.getString("nombre_tercer_titulo");
        String nominacion_tercer_titulo =
object.getString("nominacion_tercer_titulo");
        radioButtonSi2.setChecked(true);
        editTextTitulotercero.setText(t_tercer_titulo);
        editTextDenominacion3.setText(nominacion_tercer_titulo);
    }else{
        radioButtonNo2.setChecked(true);

    }
    String titulo_4 = object.getString("cuarto_titulo");
    if(titulo_4.equals("SI")){
        String t_cuarto_titulo =
object.getString("nombre_cuarto_titulo");
        String nominacion_cuarto_titulo =
object.getString("nominacion_cuarto_titulo");
        radioButtonSi3.setChecked(true);
        editTextTitulocuarto.setText(t_cuarto_titulo);
        editTextDenominacion4.setText(nominacion_cuarto_titulo);
    }else{
        radioButtonNo3.setChecked(true);

    }
}
progressDialog.dismiss();
```

DIAGRAMA DE FLUJO


Algoritmo actualizarPostgrados
Definir iGoogleSheets Como Real
Definir jsonRequest, sheet Como Cadena
Definir code Como Entero
iGoogleSheets1 ← iGNGetMethodClient[Common.BASE_URL]
response ← '?????Id=' +Common.getUsername[]+'&sheet=personas'
code ← response.code[]
code=200
jsonRequest ← ' \value\: \ + cuarto nivel + \\n'
iGoogleSheets1 ← iGoogleSheets.postStringRequestBody[jsonRequest]
cuartoNivel ← response.radioButton[]
cuartoNivel='si'
jsonRequest ← ' \value\: \ + quinto nivel + \\n'
iGoogleSheets1 ← iGoogleSheets.postStringRequestBody[jsonRequest]
quintoNivel ← response.radioButton[]
quintoNivel='si'
jsonRequest ← ' \value\: \ + tercer titulo + \\n'
iGoogleSheets1 ← iGoogleSheets.postStringRequestBody[jsonRequest]
tercerTitulo ← response.radioButton[]
tercerTitulo='si'
jsonRequest ← ' \value\: \ + cuarto titulo + \\n'
iGoogleSheets1 ← iGoogleSheets.postStringRequestBody[jsonRequest]
code ← response.radioButton[]
FinAlgoritmo

GRAFO



RUTAS

R1: 1,2,3,4,5,6,7,8,24
R2: 1,2,3,4,5,6,7,8,9,10,11,12,24
R3: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,24
R4: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,24
R4: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

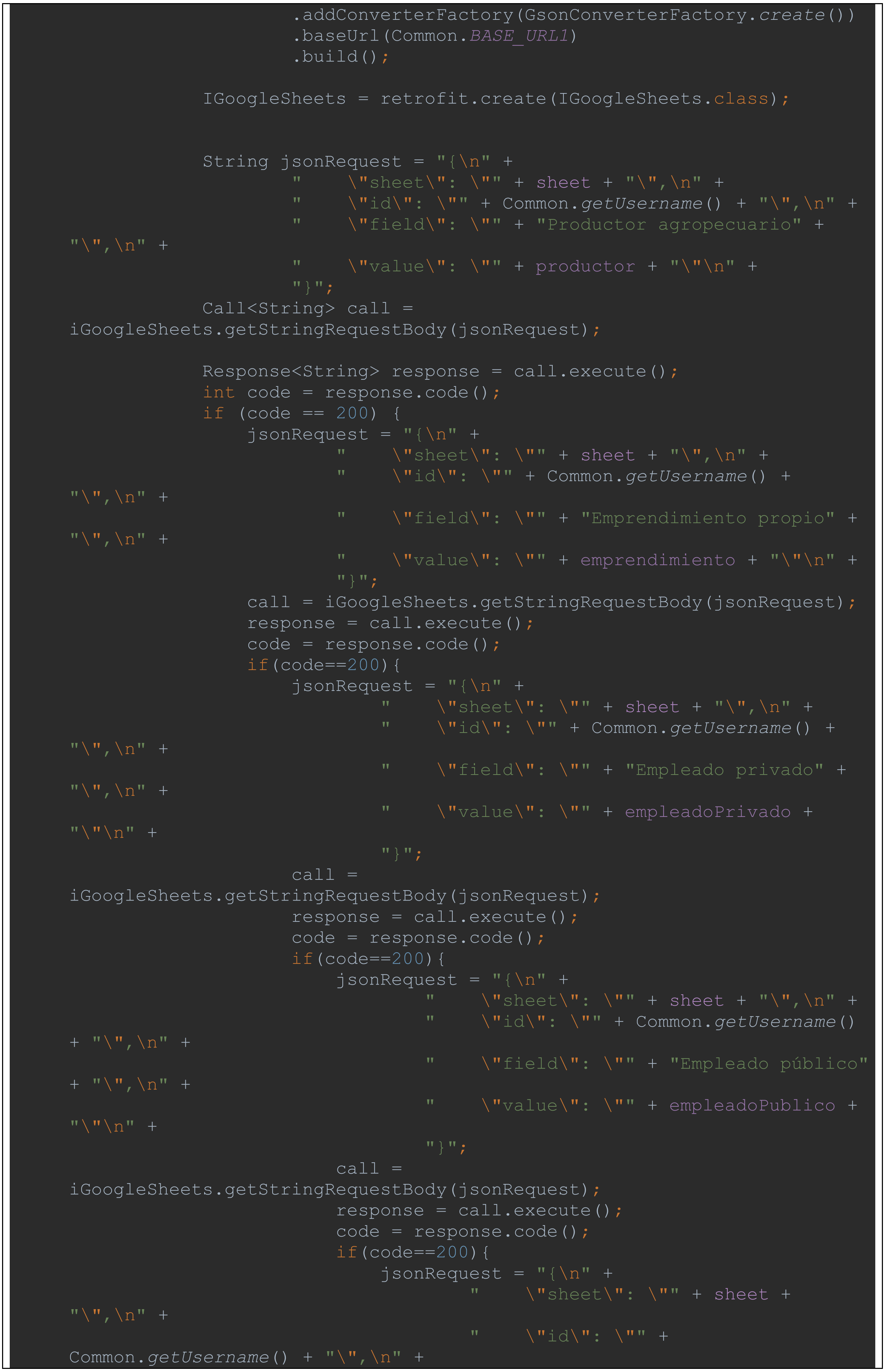
- V(G) = número de nodos predicados(decisiones)+1
  V(G)=4+1=5
- V(G) = A – N + 2
  V(G)= 17 – 14+ 2 = 5

DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

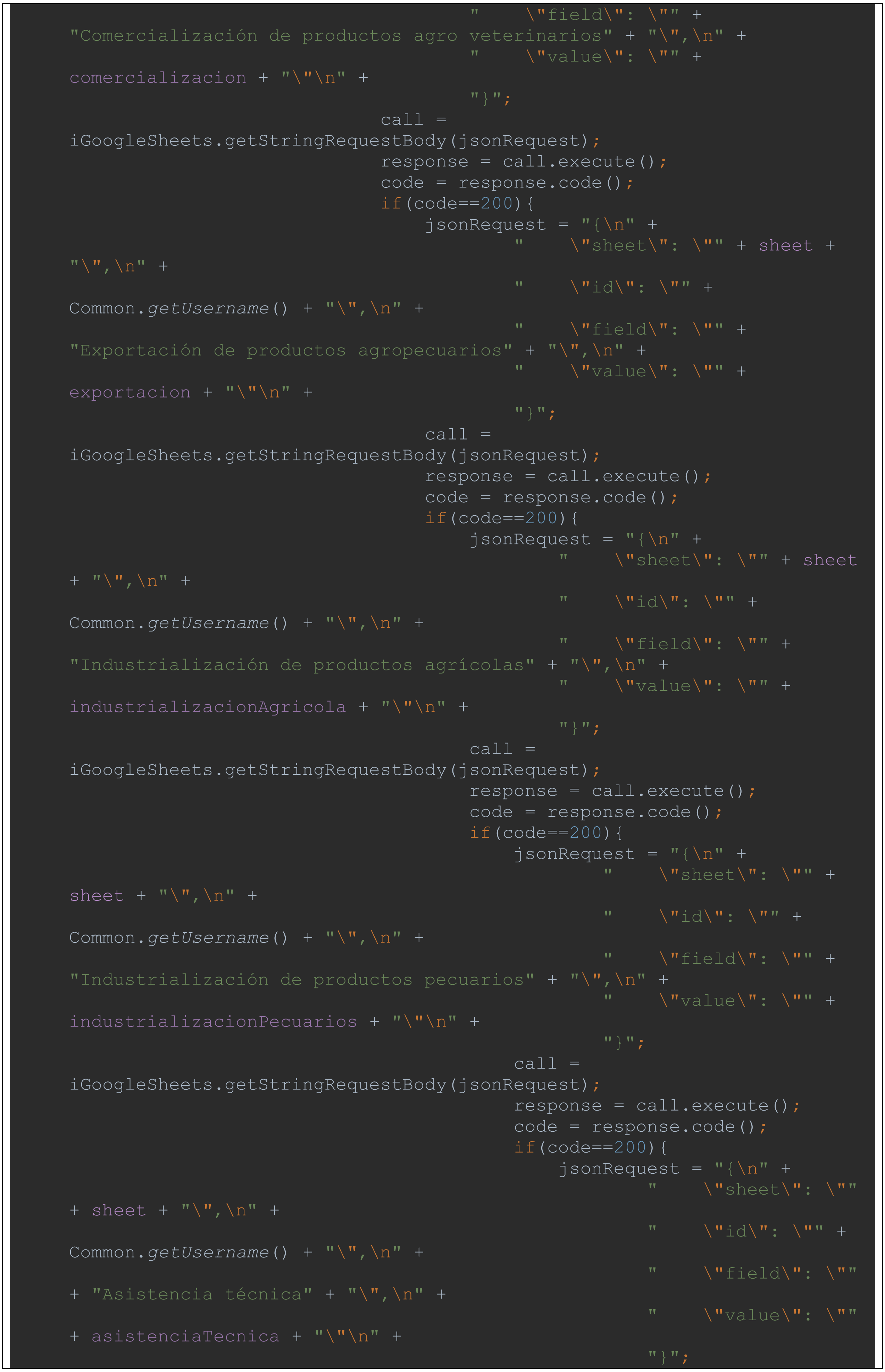
Req. 10 ACTUALIZAR INFORMACIÓN PROFESIONAL

CÓDIGO FUENTE

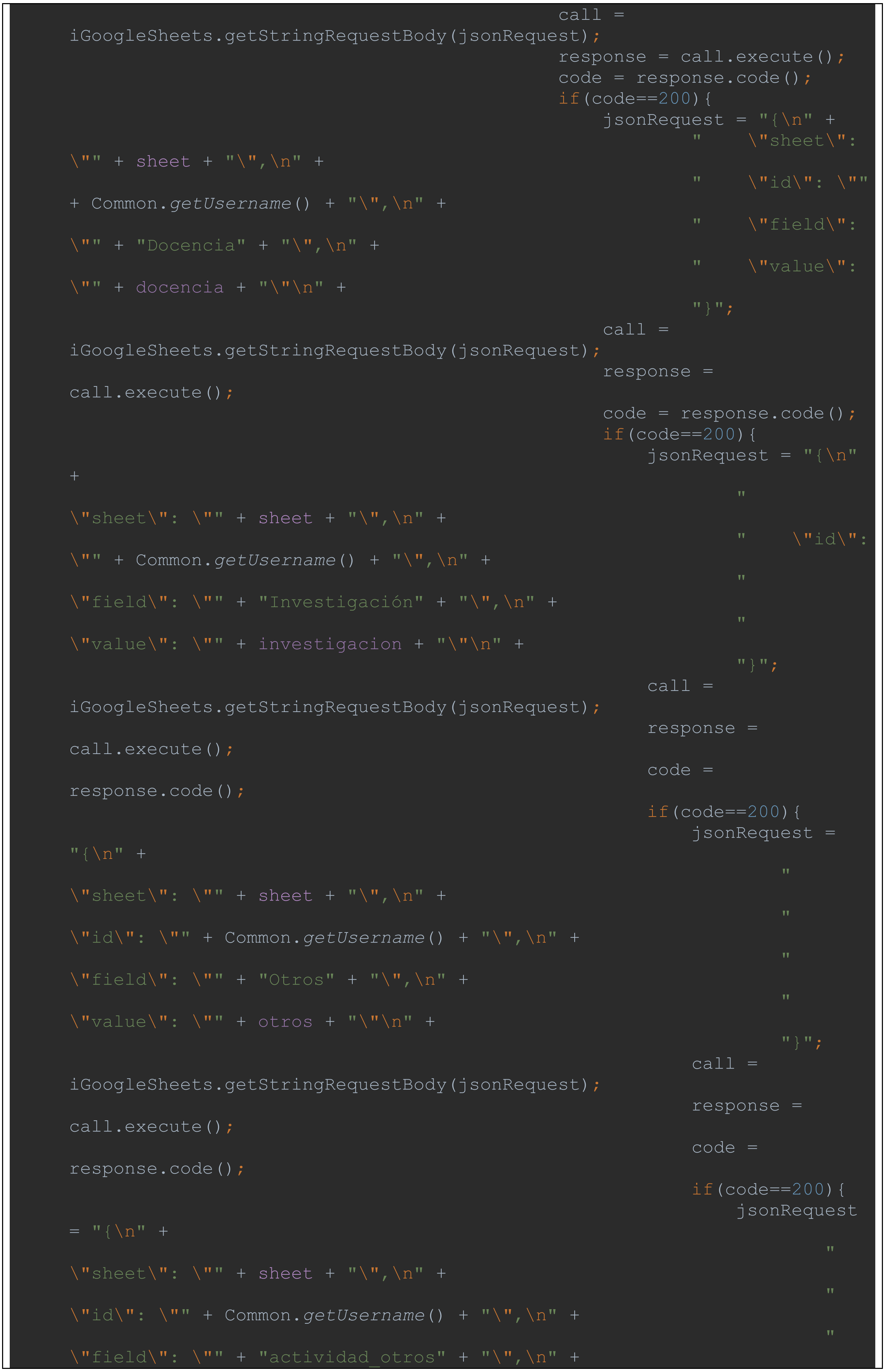
```
private void saveUpdatedData() {
         // Retrieve the updated data from the EditText fields
         ProgressDialog =
     ProgressDialog.show(FormularioActivityProfesional.this,
                 "Actualizando datos",
                 "Espere por favor",
                 true,
                 false);


         if(checkBoxProductor.isChecked()){
             productor="SI";
         }
         if(checkBoxEmprendimiento.isChecked()){
             emprendimiento="SI";
         }
         if(checkBoxEmpleadoPublico.isChecked()){
             empleadoPublico="SI";
         }
         if(checkBoxEmpleadoPrivado.isChecked()){
             empleadoPrivado="SI";
         }
         if(checkBoxComercializacion.isChecked()){
             comercializacion="SI";
         }

         if(checkBoxExportacion.isChecked()){
             exportacion="SI";
         }
         if(checkBoxIndustrializacionAgricola.isChecked()){
             industrializacionAgricola="SI";
         }
         if(checkBoxIndustrializacionPecuarios.isChecked()){
             industrializacionPecuarios="SI";
         }
         if(checkBoxAsistenciaTecnica.isChecked()){
             asistenciaTecnica="SI";
         }
         if(checkBoxDocencia.isChecked()){
             docencia="SI";
         }
         if(checkBoxInvestigacion.isChecked()){
             investigacion="SI";
         }
         if(checkBoxOtros.isChecked()){
             otros="SI";
             actividadotros=editTextOtros.getText().toString();
         }


         String sheet="actividad_profesional";




         AsyncTask.execute(() -> {
             try {
                 Retrofit = new Retrofit.Builder()

     .addConverterFactory(ScalarsConverterFactory.create())
```



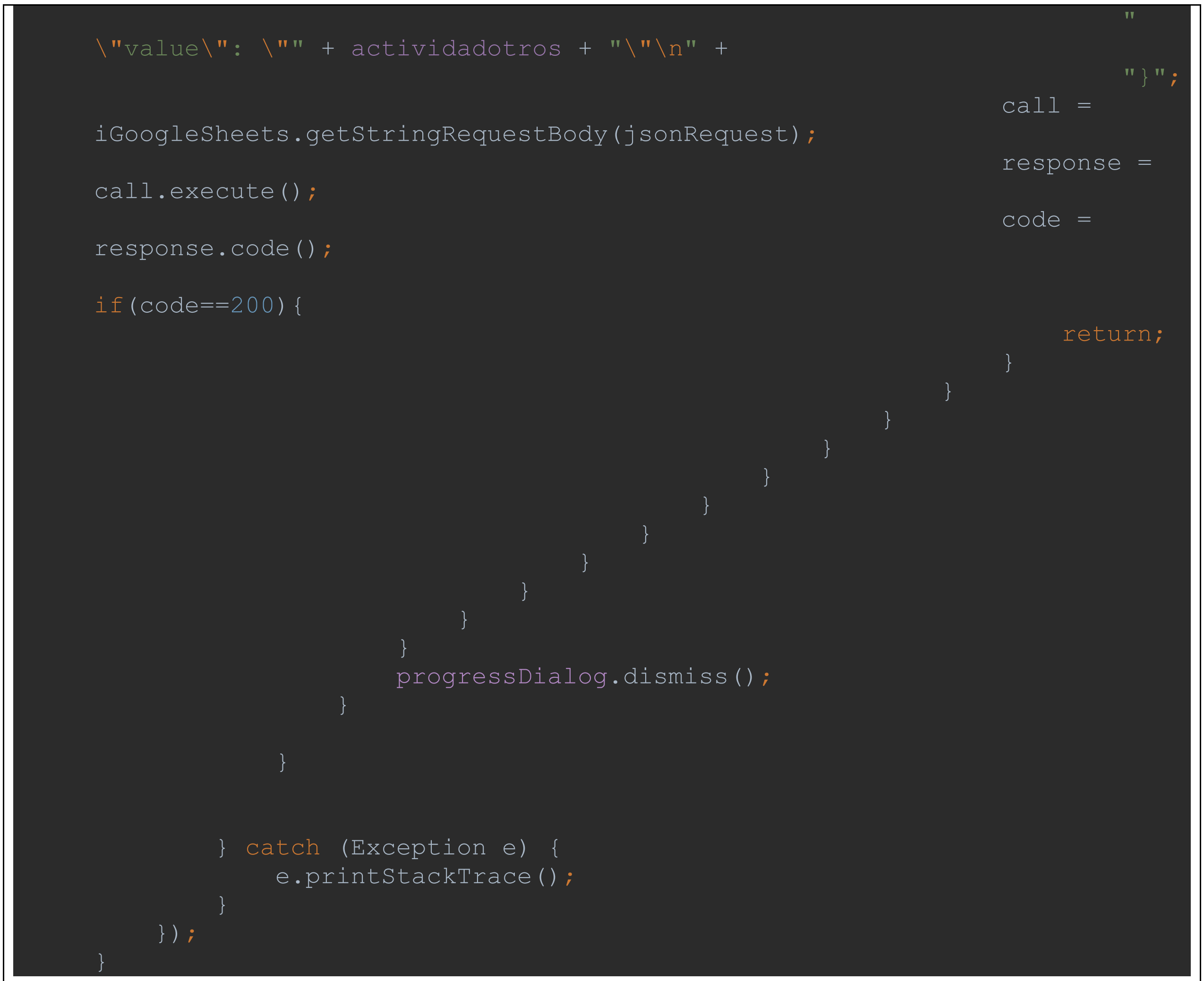
```
                    .addConverterFactory(GsonConverterFactory.create())
                    .baseUrl(Common.BASE_URL1)
                    .build();

            IGoogleSheets = retrofit.create(IGoogleSheets.class);


            String jsonRequest = "{\n" +
                    "    \"sheet\": \"" + sheet + "\",\n" +
                    "    \"id\": \"" + Common.getUsername() + "\",\n" +
                    "    \"field\": \"" + "Productor agropecuario" +
"\",\n" +
                    "    \"value\": \"" + productor + "\"\n" +
                    "}";
            Call<String> call =
iGoogleSheets.getStringRequestBody(jsonRequest);

            Response<String> response = call.execute();
            int code = response.code();
            if (code == 200) {
                jsonRequest = "{\n" +
                        "    \"sheet\": \"" + sheet + "\",\n" +
                        "    \"id\": \"" + Common.getUsername() +
"\",\n" +
                        "    \"field\": \"" + "Emprendimiento propio" +
"\",\n" +
                        "    \"value\": \"" + emprendimiento + "\"\n" +
                        "}";
                call = iGoogleSheets.getStringRequestBody(jsonRequest);
                response = call.execute();
                code = response.code();
                if(code==200){
                    jsonRequest = "{\n" +
                            "    \"sheet\": \"" + sheet + "\",\n" +
                            "    \"id\": \"" + Common.getUsername() +
"\",\n" +
                            "    \"field\": \"" + "Empleado privado" +
"\",\n" +
                            "    \"value\": \"" + empleadoPrivado +
"\"\n" +
                            "}";
                    call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                    response = call.execute();
                    code = response.code();
                    if(code==200){
                        jsonRequest = "{\n" +
                                "    \"sheet\": \"" + sheet + "\",\n" +
                                "    \"id\": \"" + Common.getUsername()
+ "\",\n" +
                                "    \"field\": \"" + "Empleado público"
+ "\",\n" +
                                "    \"value\": \"" + empleadoPublico +
"\"\n" +
                                "}";
                        call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                        response = call.execute();
                        code = response.code();
                        if(code==200){
                            jsonRequest = "{\n" +
                                    "    \"sheet\": \"" + sheet +
"\",\n" +
                                    "    \"id\": \"" +
Common.getUsername() + "\",\n" +
```



```
                                    "    \"field\": \"" +
"Comercialización de productos agro veterinarios" + "\",\n" +
                                    "    \"value\": \"" +
comercializacion + "\"\n" +
                                    "}";
                            call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                            response = call.execute();
                            code = response.code();
                            if(code==200){
                                jsonRequest = "{\n" +
                                        "    \"sheet\": \"" + sheet +
"\",\n" +
                                        "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                        "    \"field\": \"" +
"Exportación de productos agropecuarios" + "\",\n" +
                                        "    \"value\": \"" +
exportacion + "\"\n" +
                                        "}";
                                call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                response = call.execute();
                                code = response.code();
                                if(code==200){
                                    jsonRequest = "{\n" +
                                            "    \"sheet\": \"" + sheet
+ "\",\n" +
                                            "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                            "    \"field\": \"" +
"Industrialización de productos agrícolas" + "\",\n" +
                                            "    \"value\": \"" +
industrializacionAgricola + "\"\n" +
                                            "}";
                                    call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                    response = call.execute();
                                    code = response.code();
                                    if(code==200){
                                        jsonRequest = "{\n" +
                                                "    \"sheet\": \"" +
sheet + "\",\n" +
                                                "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                                "    \"field\": \"" +
"Industrialización de productos pecuarios" + "\",\n" +
                                                "    \"value\": \"" +
industrializacionPecuarios + "\"\n" +
                                                "}";
                                        call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                        response = call.execute();
                                        code = response.code();
                                        if(code==200){
                                            jsonRequest = "{\n" +
                                                    "    \"sheet\": \""
+ sheet + "\",\n" +
                                                    "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                                    "    \"field\": \""
+ "Asistencia técnica" + "\",\n" +
                                                    "    \"value\": \""
+ asistenciaTecnica + "\"\n" +
                                                    "}";
```

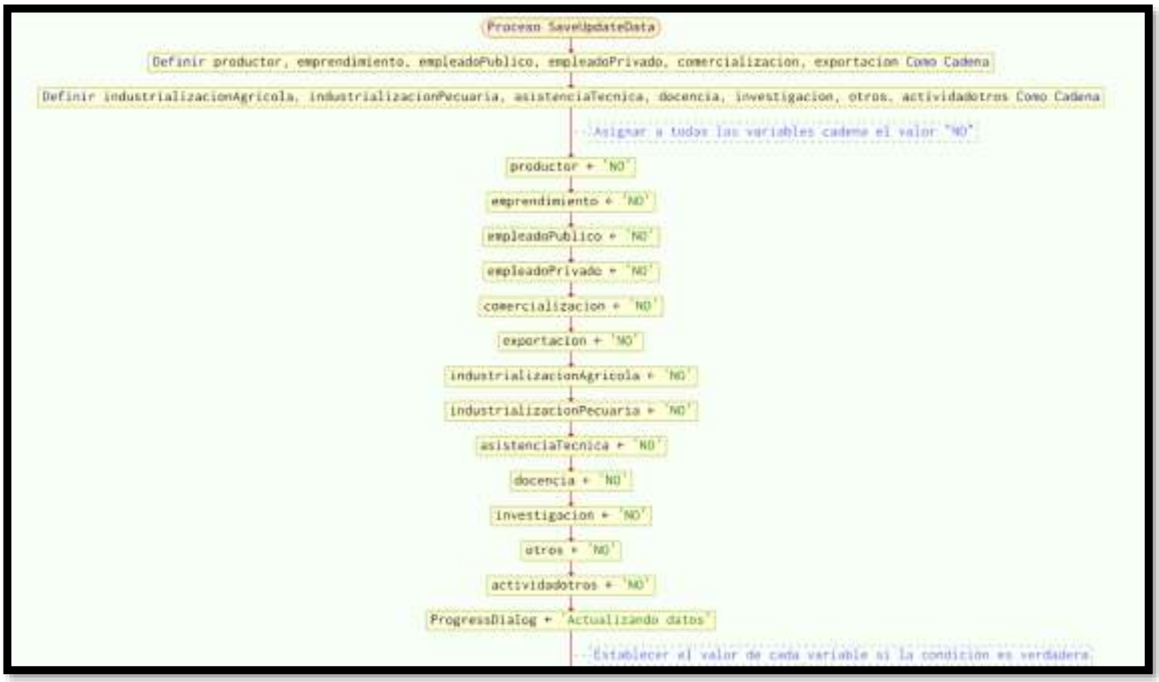


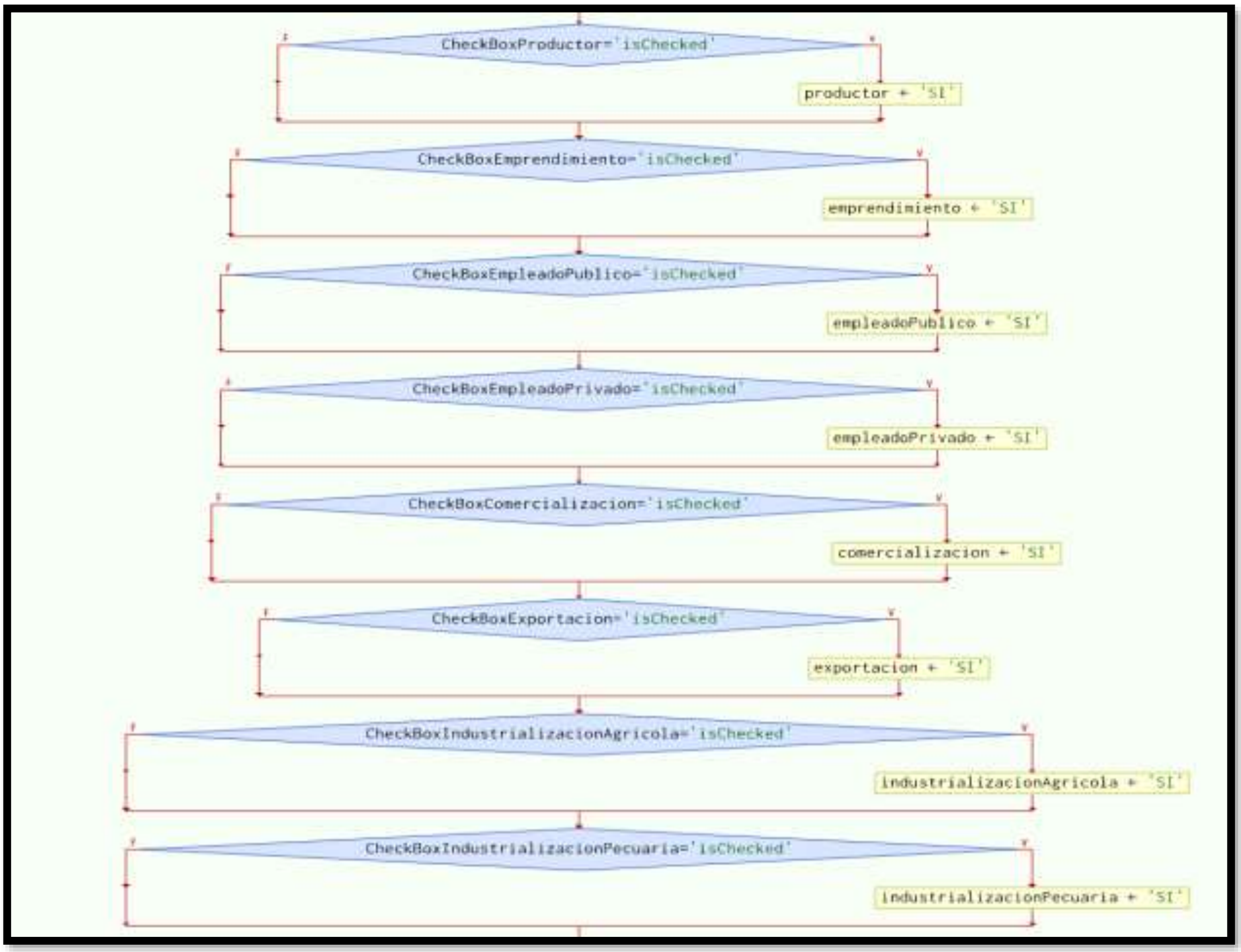
```
                                            call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                            response = call.execute();
                                            code = response.code();
                                            if(code==200){
                                                jsonRequest = "{\n" +
                                                        "    \"sheet\":
\"" + sheet + "\",\n" +
                                                        "    \"id\": \""
+ Common.getUsername() + "\",\n" +
                                                        "    \"field\":
\"" + "Docencia" + "\",\n" +
                                                        "    \"value\":
\"" + docencia + "\"\n" +
                                                        "}";
                                                call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                                response =
call.execute();
                                                code = response.code();
                                                if(code==200){
                                                    jsonRequest = "{\n"
+
                                                            "
\"sheet\": \"" + sheet + "\",\n" +
                                                            "    \"id\":
\"" + Common.getUsername() + "\",\n" +
                                                            "
\"field\": \"" + "Investigación" + "\",\n" +
                                                            "
\"value\": \"" + investigacion + "\"\n" +
                                                            "}";
                                                    call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                                    response =
call.execute();
                                                    code =
response.code();
                                                    if(code==200){
                                                        jsonRequest =
"{\n" +
                                                                "
\"sheet\": \"" + sheet + "\",\n" +
                                                                "
\"id\": \"" + Common.getUsername() + "\",\n" +
                                                                "
\"field\": \"" + "Otros" + "\",\n" +
                                                                "
\"value\": \"" + otros + "\"\n" +
                                                                "}";
                                                        call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                                        response =
call.execute();
                                                        code =
response.code();
                                                        if(code==200){
                                                            jsonRequest
= "{\n" +
                                                                    "
\"sheet\": \"" + sheet + "\",\n" +
                                                                    "
\"id\": \"" + Common.getUsername() + "\",\n" +
                                                                    "
\"field\": \"" + "actividad_otros" + "\",\n" +
```

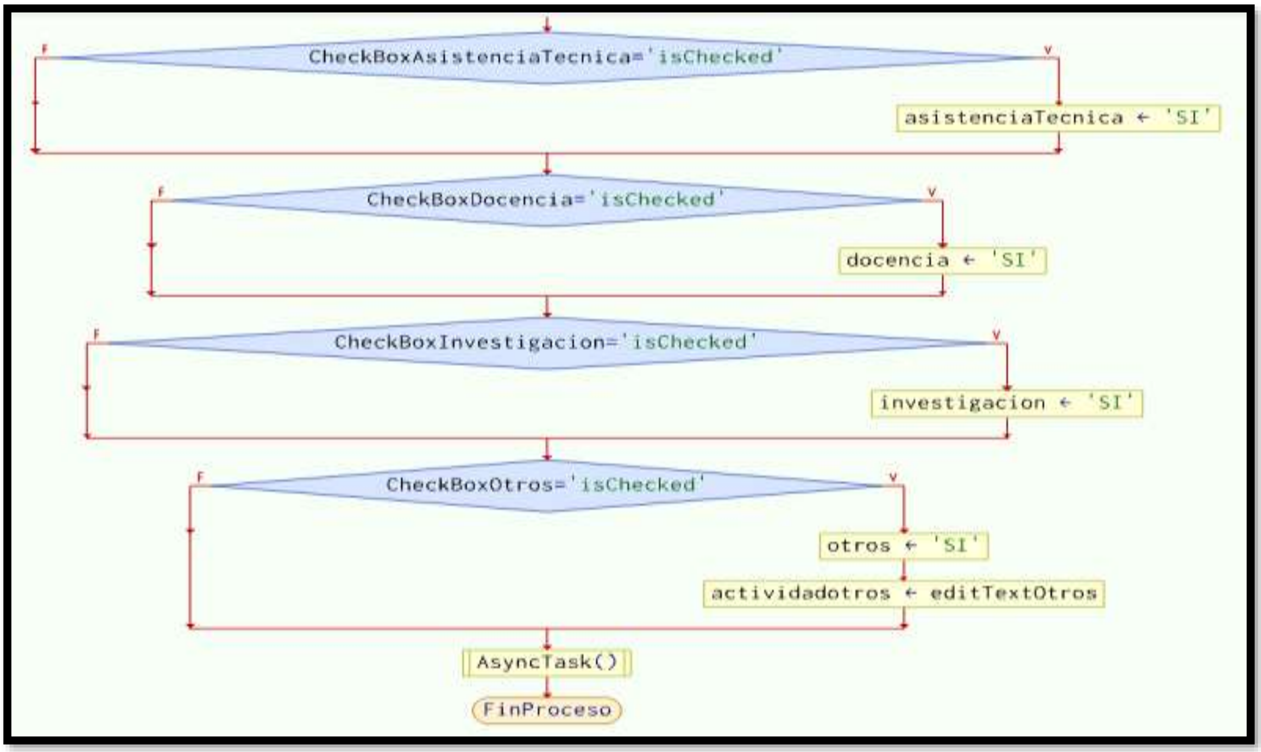


```
                                                                     "
\"value\": \"" + actividadotros + "\"\n" +
                                                                    "}";
                                                            call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                                            response =
call.execute();
                                                            code =
response.code();

if(code==200){
                                                                return;
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                    progressDialog.dismiss();
                }

            }


        } catch (Exception e) {
            e.printStackTrace();
        }
    });
}
```

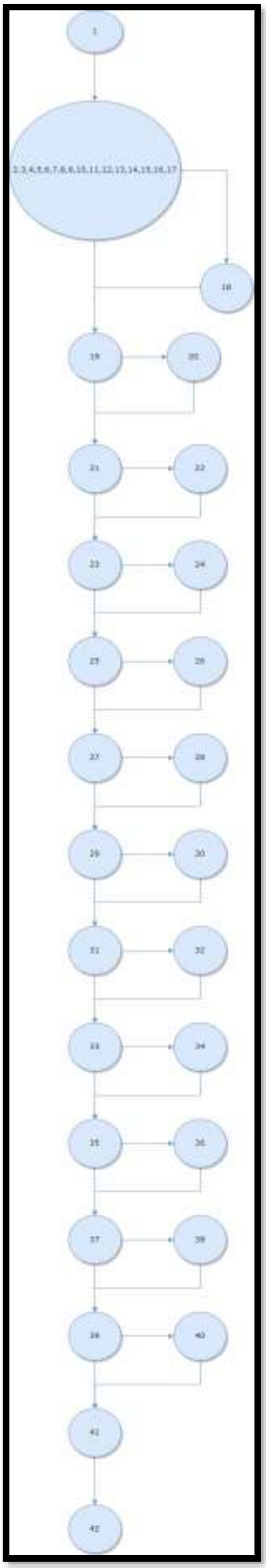
DIAGRAMA DE FLUJO


Proceso SaveUpdateData
Definir productor, emprendimiento, empleadoPublico, empleadoPrivado, comercializacion, exportacion Como Cadena
Definir industrializacionAgricola, industrializacionPecuaria, asistenciaTecnica, docencia, investigacion, otros, actividadotros Como Cadena
Asignar a todas las variables cadena el valor "NO"
productor ← 'NO'
emprendimiento ← 'NO'
empleadoPublico ← 'NO'
empleadoPrivado ← 'NO'
comercializacion ← 'NO'
exportacion ← 'NO'
industrializacionAgricola ← 'NO'
industrializacionPecuaria ← 'NO'
asistenciaTecnica ← 'NO'
docencia ← 'NO'
investigacion ← 'NO'
otros ← 'NO'
actividadotros ← 'NO'
ProgressDialog ← 'Actualizando datos'
Establecer el valor de cada variable si la condicion es verdadera


F
CheckBoxProductor='isChecked'
V
productor ← 'SI'
F
CheckBoxEmprendimiento='isChecked'
V
emprendimiento ← 'SI'
F
CheckBoxEmpleadoPublico='isChecked'
V
empleadoPublico ← 'SI'
F
CheckBoxEmpleadoPrivado='isChecked'
V
empleadoPrivado ← 'SI'
F
CheckBoxComercializacion='isChecked'
V
comercializacion ← 'SI'
F
CheckBoxExportacion='isChecked'
V
exportacion ← 'SI'
F
CheckBoxIndustrializacionAgricola='isChecked'
V
industrializacionAgricola ← 'SI'
F
CheckBoxIndustrializacionPecuaria='isChecked'
V
industrializacionPecuaria ← 'SI'


F
CheckBoxAsistenciaTecnica='isChecked'
V
asistenciaTecnica ← 'SI'
F
CheckBoxDocencia='isChecked'
V
docencia ← 'SI'
F
CheckBoxInvestigacion='isChecked'
V
investigacion ← 'SI'
F
CheckBoxOtros='isChecked'
V
otros ← 'SI'
actividadotros ← editTextOtros
AsyncTask()
FinProceso

GRAFO


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
32
33
34
35
36
37
38
39
40
41
42

RUTAS

R1: 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, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R2: 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, 32, 33, 34, 35, 36, 37, 38, 39, 41, 42
R3: 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, 32, 33, 34, 35, 36, 37, 39, 40, 41, 42
R4: 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, 32, 33, 34, 35, 37, 38, 39, 40, 41, 42
R5: 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, 32, 33, 35, 36, 37, 38, 39, 40, 41, 42
R6: 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, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R7: 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, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R8: 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, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R9: 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, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R10: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R11: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R12: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R13: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R14: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 42

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

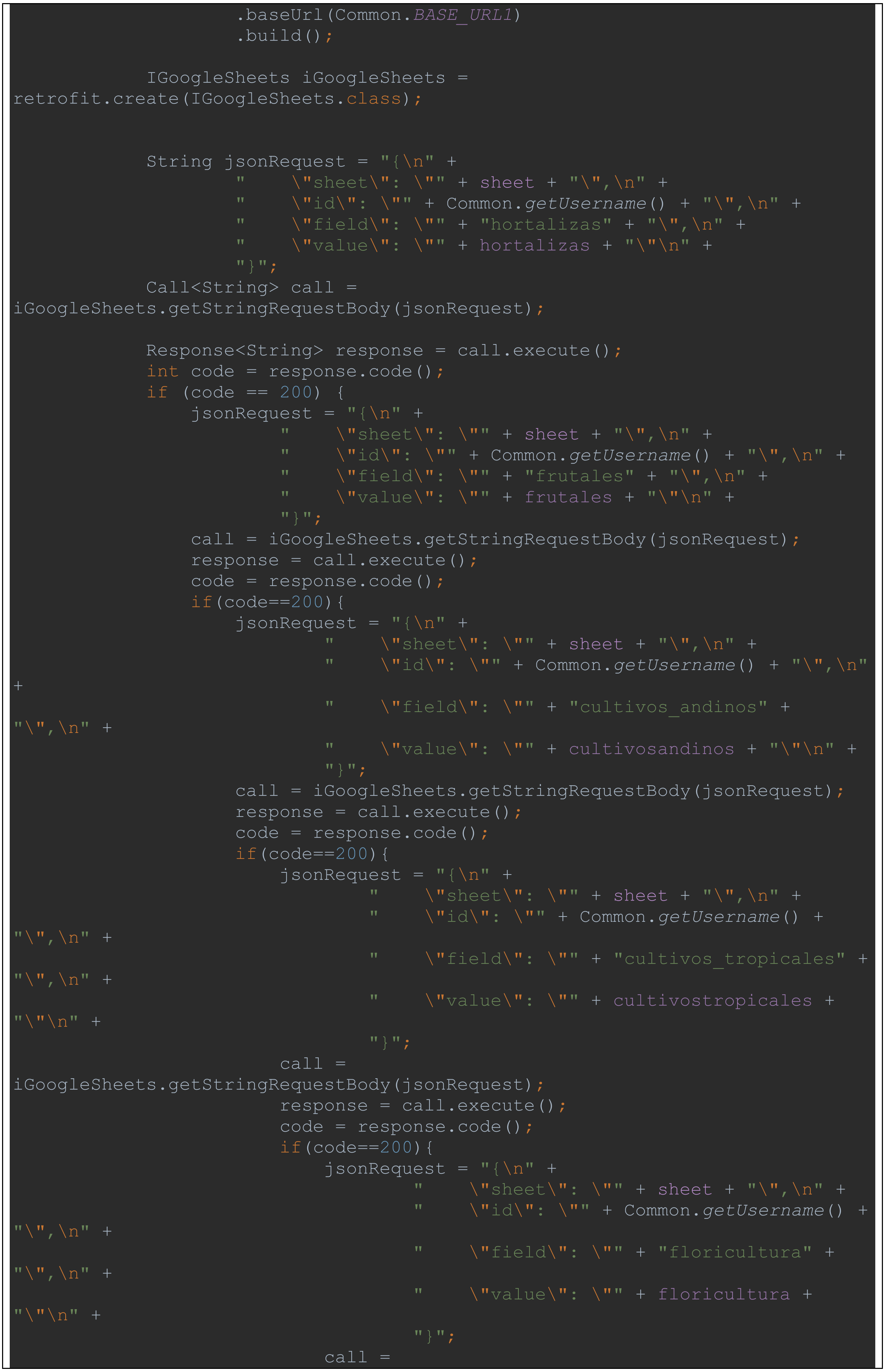
- V(G) = número de nodos predicados(decisiones)+1
  V(G)=12+1=13
- V(G) = A – N + 2
  V(G)= 38– 27 + 2 = 13

DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

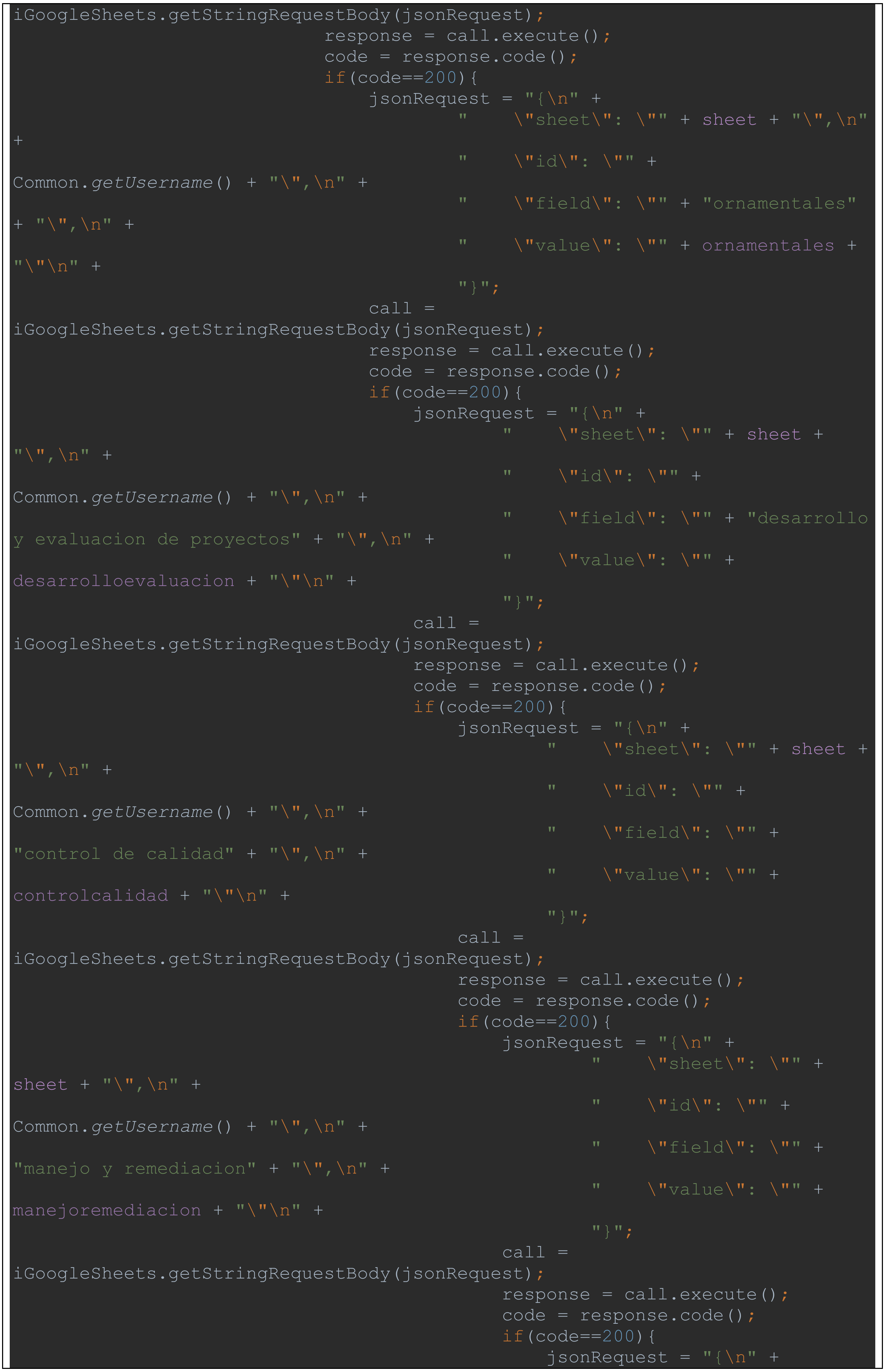
Req. 11 ACTUALIZAR ACTIVIDAD AGRÍCOLA

CÓDIGO FUENTE

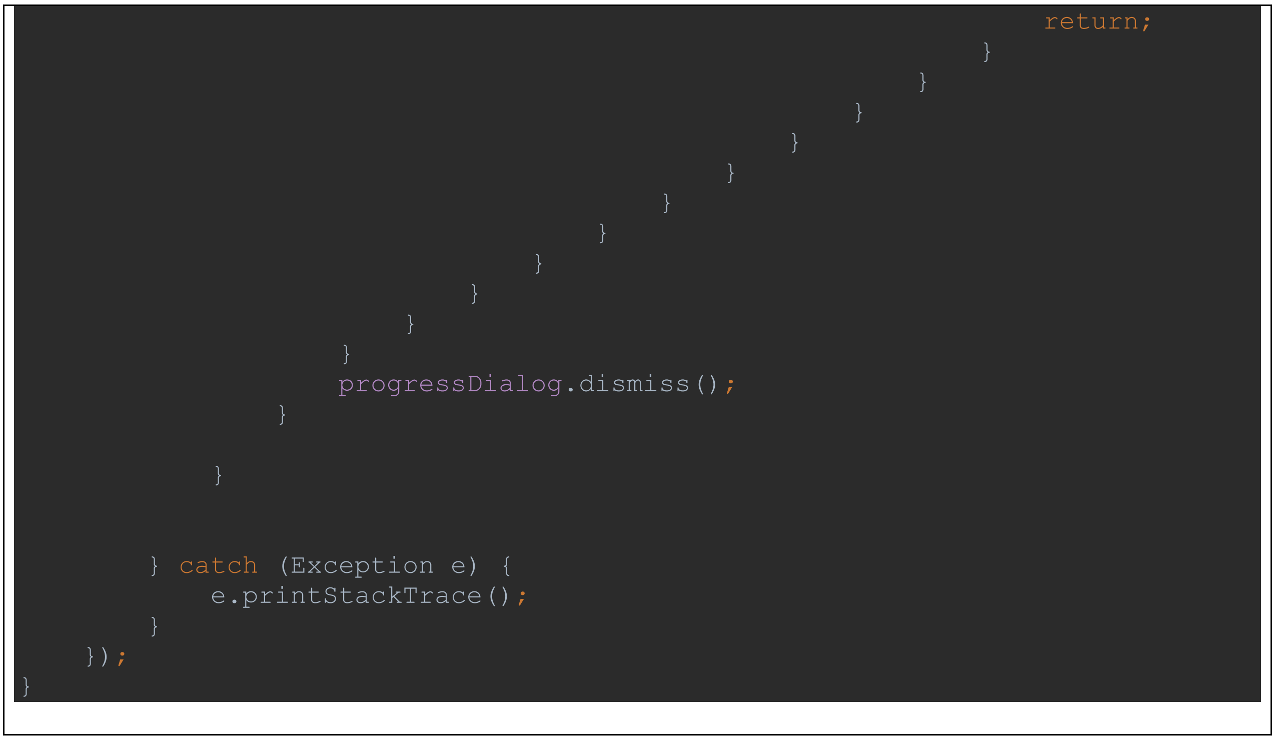
```
private void saveUpdatedData() {
    // Retrieve the updated data from the EditText fields
    ProgressDialog progressDialog =
ProgressDialog.show(FormularioActivityAgricola.this,
            "Actualizando datos",
            "Espere por favor",
            true,
            false);


    if(checkBoxHortalizas.isChecked()){
        hortalizas="SI";
    }
    if(checkBoxFrutales.isChecked()){
        frutales="SI";
    }
    if(checkBoxCultivosAndinos.isChecked()){
        cultivosandinos="SI";
    }
    if(checkBoxCultivosTropicales.isChecked()){
        cultivostropicales="SI";
    }
    if(checkBoxFloricultura.isChecked()){
        floricultura="SI";
    }

    if(checkBoxOrnamentales.isChecked()){
        ornamentales="SI";
    }
    if(checkBoxDesarrolloEvaluacion.isChecked()){
        desarrolloevaluacion="SI";
    }
    if(checkBoxControlCalidad.isChecked()){
        controlcalidad="SI";
    }
    if(checkBoxManejoRemediacion.isChecked()){
        manejoremediacion="SI";
    }
    if(checkBoxAgroecologiaMedioambiente.isChecked()){
        agroecologia="SI";
    }
    if(checkBoxNutricionVegetal.isChecked()){
        nuticionvegetal="SI";
    }
    if(checkBoxOtros.isChecked()){
        otros="SI";
        actividadotros=editTextOtros.getText().toString();
    }


    String sheet="area_actividad_agricola";




    AsyncTask.execute(() -> {
        try {
            Retrofit retrofit = new Retrofit.Builder()
                    .addConverterFactory(ScalarsConverterFactory.create())
                    .addConverterFactory(GsonConverterFactory.create())
```



```
                    .baseUrl(Common.BASE_URL1)
                    .build();

            IGoogleSheets iGoogleSheets =
retrofit.create(IGoogleSheets.class);


            String jsonRequest = "{\n" +
                    "    \"sheet\": \"" + sheet + "\",\n" +
                    "    \"id\": \"" + Common.getUsername() + "\",\n" +
                    "    \"field\": \"" + "hortalizas" + "\",\n" +
                    "    \"value\": \"" + hortalizas + "\"\n" +
                    "}";
            Call<String> call =
iGoogleSheets.getStringRequestBody(jsonRequest);

            Response<String> response = call.execute();
            int code = response.code();
            if (code == 200) {
                jsonRequest = "{\n" +
                        "    \"sheet\": \"" + sheet + "\",\n" +
                        "    \"id\": \"" + Common.getUsername() + "\",\n" +
                        "    \"field\": \"" + "frutales" + "\",\n" +
                        "    \"value\": \"" + frutales + "\"\n" +
                        "}";
                call = iGoogleSheets.getStringRequestBody(jsonRequest);
                response = call.execute();
                code = response.code();
                if(code==200){
                    jsonRequest = "{\n" +
                            "    \"sheet\": \"" + sheet + "\",\n" +
                            "    \"id\": \"" + Common.getUsername() + "\",\n"
+
                            "    \"field\": \"" + "cultivos_andinos" +
"\",\n" +
                            "    \"value\": \"" + cultivosandinos + "\"\n" +
                            "}";
                    call = iGoogleSheets.getStringRequestBody(jsonRequest);
                    response = call.execute();
                    code = response.code();
                    if(code==200){
                        jsonRequest = "{\n" +
                                "    \"sheet\": \"" + sheet + "\",\n" +
                                "    \"id\": \"" + Common.getUsername() +
"\",\n" +
                                "    \"field\": \"" + "cultivos_tropicales" +
"\",\n" +
                                "    \"value\": \"" + cultivostropicales +
"\"\n" +
                                "}";
                        call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                        response = call.execute();
                        code = response.code();
                        if(code==200){
                            jsonRequest = "{\n" +
                                    "    \"sheet\": \"" + sheet + "\",\n" +
                                    "    \"id\": \"" + Common.getUsername() +
"\",\n" +
                                    "    \"field\": \"" + "floricultura" +
"\",\n" +
                                    "    \"value\": \"" + floricultura +
"\"\n" +
                                    "}";
                            call =
```

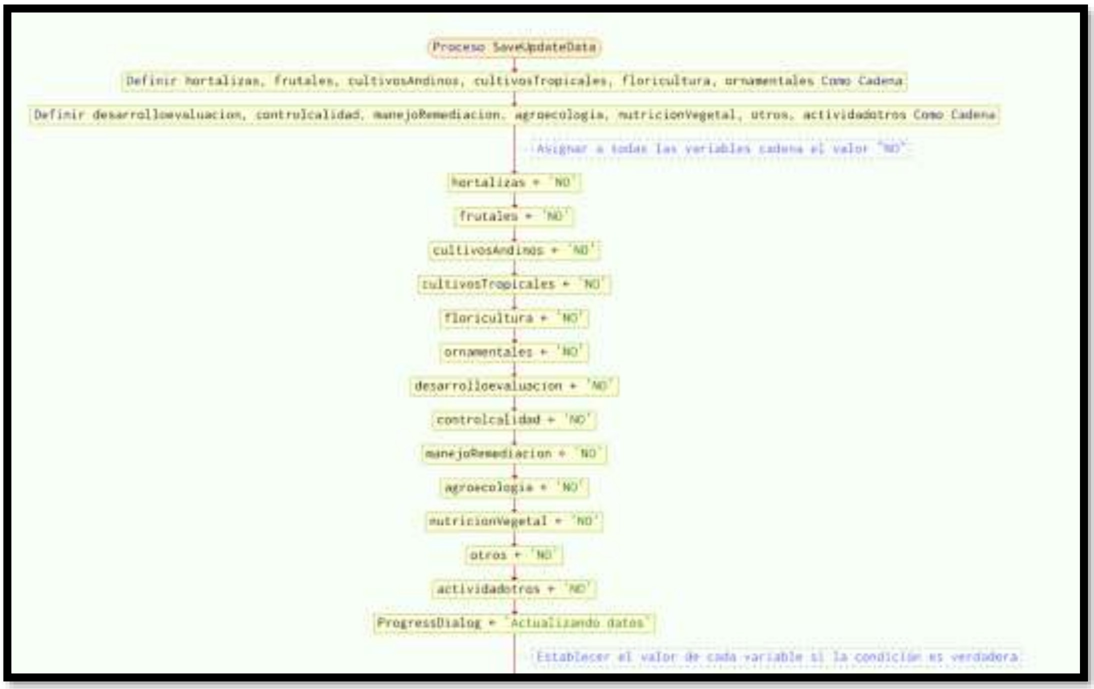


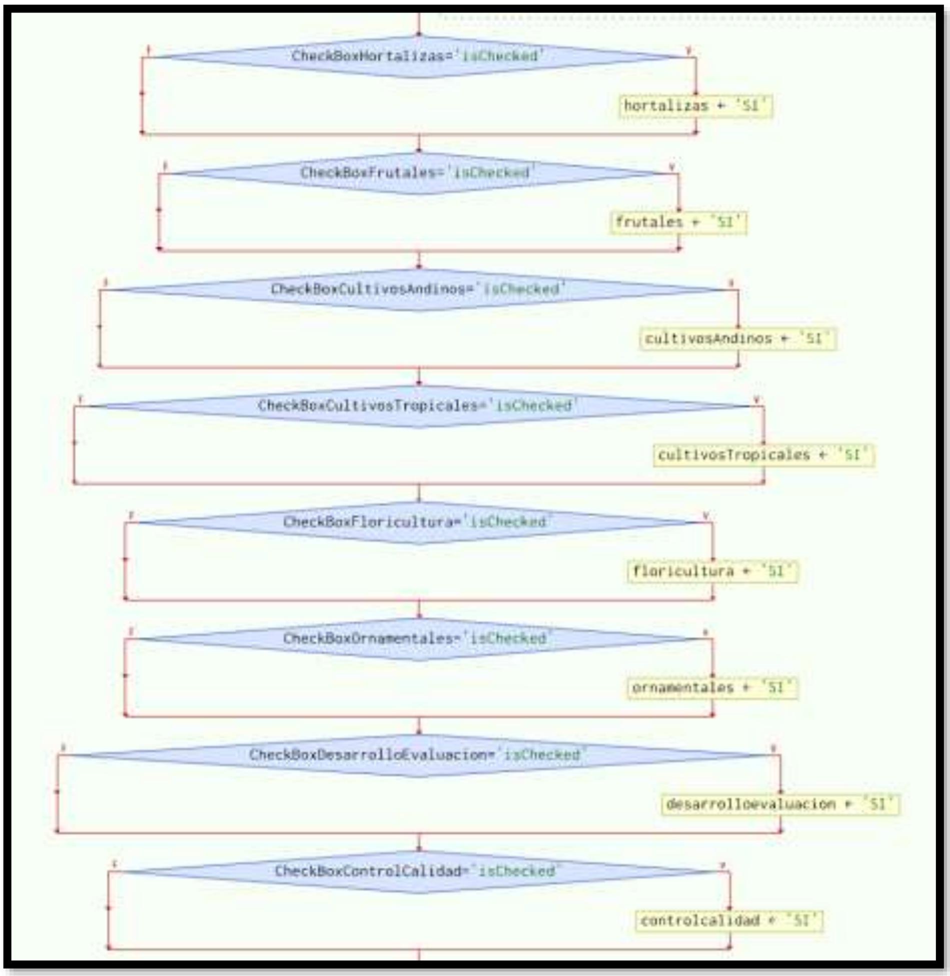
```
iGoogleSheets.getStringRequestBody(jsonRequest);
                            response = call.execute();
                            code = response.code();
                            if(code==200){
                                jsonRequest = "{\n" +
                                        "    \"sheet\": \"" + sheet + "\",\n"
+
                                        "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                        "    \"field\": \"" + "ornamentales"
+ "\",\n" +
                                        "    \"value\": \"" + ornamentales +
"\"\n" +
                                        "}";
                                call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                response = call.execute();
                                code = response.code();
                                if(code==200){
                                    jsonRequest = "{\n" +
                                            "    \"sheet\": \"" + sheet +
"\",\n" +
                                            "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                            "    \"field\": \"" + "desarrollo
y evaluacion de proyectos" + "\",\n" +
                                            "    \"value\": \"" +
desarrolloevaluacion + "\"\n" +
                                            "}";
                                    call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                    response = call.execute();
                                    code = response.code();
                                    if(code==200){
                                        jsonRequest = "{\n" +
                                                "    \"sheet\": \"" + sheet +
"\",\n" +
                                                "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                                "    \"field\": \"" +
"control de calidad" + "\",\n" +
                                                "    \"value\": \"" +
controlcalidad + "\"\n" +
                                                "}";
                                        call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                        response = call.execute();
                                        code = response.code();
                                        if(code==200){
                                            jsonRequest = "{\n" +
                                                    "    \"sheet\": \"" +
sheet + "\",\n" +
                                                    "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                                    "    \"field\": \"" +
"manejo y remediacion" + "\",\n" +
                                                    "    \"value\": \"" +
manejoremediacion + "\"\n" +
                                                    "}";
                                            call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                            response = call.execute();
                                            code = response.code();
                                            if(code==200){
                                                jsonRequest = "{\n" +
                                                        "    \"sheet\": \"" +
sheet + "\",\n" +
                                                        "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                                        "    \"field\": \"" +
"agroecologia y medioambiente" + "\",\n" +
                                                        "    \"value\": \"" +
agroecologia + "\"\n" +
                                                        "}";
                                                call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                                response = call.execute();
                                                code = response.code();
                                                if(code==200){
                                                    jsonRequest = "{\n" +
                                                            "    \"sheet\":
\"" + sheet + "\",\n" +
                                                            "    \"id\": \""
+ Common.getUsername() + "\",\n" +
                                                            "    \"field\":
\"" + "nutricion vegetal" + "\",\n" +
                                                            "    \"value\":
\"" + nuticionvegetal + "\"\n" +
                                                            "}";
                                                    call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                                    response =
call.execute();
                                                    code = response.code();
                                                    if(code==200){
                                                        jsonRequest = "{\n" +
                                                                "
\"sheet\": \"" + sheet + "\",\n" +
                                                                "    \"id\":
\"" + Common.getUsername() + "\",\n" +
                                                                "
\"field\": \"" + "otros" + "\",\n" +
                                                                "
\"value\": \"" + otros + "\"\n" +
                                                                "}";
                                                        call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                                        response =
call.execute();
                                                        code =
response.code();
                                                        if(code==200){
                                                            jsonRequest =
"{\n" +
                                                                    "
\"sheet\": \"" + sheet + "\",\n" +
                                                                    "
\"id\": \"" + Common.getUsername() + "\",\n" +
                                                                    "
\"field\": \"" + "area_otros" + "\",\n" +
                                                                    "
\"value\": \"" + actividadotros + "\"\n" +
                                                                    "}";
                                                            call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                                            response =
call.execute();
                                                            code =
response.code();
                                                            if(code==200){
```

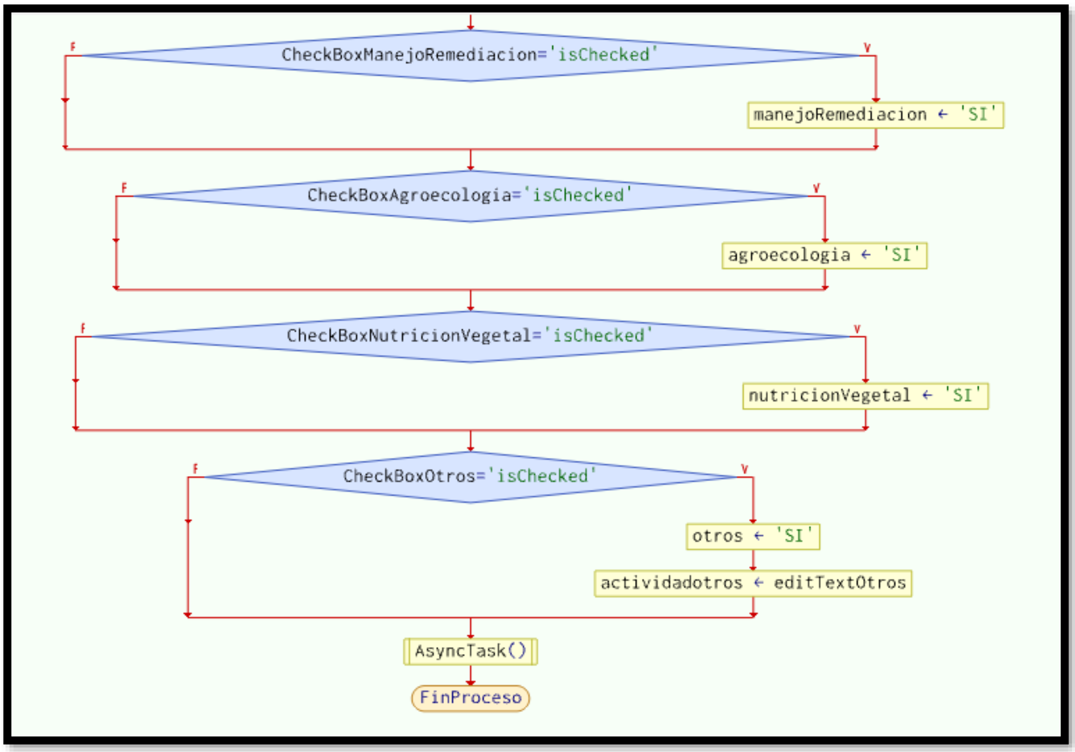


```
                                                                return;
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                    progressDialog.dismiss();
                }

            }


        } catch (Exception e) {
            e.printStackTrace();
        }
    });
}
```

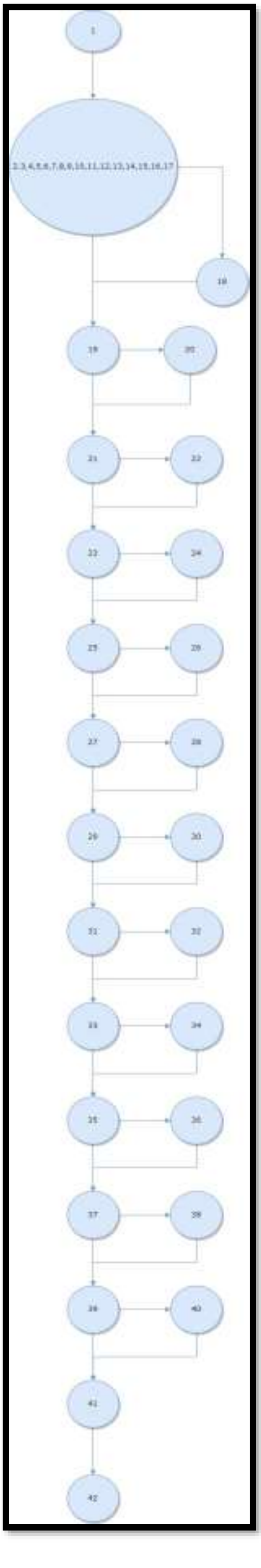
DIAGRAMA DE FLUJO


Proceso SaveUpdateData
Definir hortalizas, frutales, cultivosAndinos, cultivosTropicales, floricultura, ornamentales Como Cadena
Definir desarrolloevaluacion, controlcalidad, manejoRemediacion, agroecologia, nutricionVegetal, otros, actividadotros Como Cadena
Asignar a todas las variables cadena el valor "NO"
hortalizas ← 'NO'
frutales ← 'NO'
cultivosAndinos ← 'NO'
cultivosTropicales ← 'NO'
floricultura ← 'NO'
ornamentales ← 'NO'
desarrolloevaluacion ← 'NO'
controlcalidad ← 'NO'
manejoRemediacion ← 'NO'
agroecologia ← 'NO'
nutricionVegetal ← 'NO'
otros ← 'NO'
actividadotros ← 'NO'
ProgressDialog ← 'Actualizando datos'
Establecer el valor de cada variable si la condición es verdadera


F
CheckBoxHortalizas='isChecked'
V
hortalizas ← 'SI'
F
CheckBoxFrutales='isChecked'
V
frutales ← 'SI'
F
CheckBoxCultivosAndinos='isChecked'
V
cultivosAndinos ← 'SI'
F
CheckBoxCultivosTropicales='isChecked'
V
cultivosTropicales ← 'SI'
F
CheckBoxFloricultura='isChecked'
V
floricultura ← 'SI'
F
CheckBoxOrnamentales='isChecked'
V
ornamentales ← 'SI'
F
CheckBoxDesarrolloEvaluacion='isChecked'
V
desarrolloevaluacion ← 'SI'
F
CheckBoxControlCalidad='isChecked'
V
controlcalidad ← 'SI'


F
CheckBoxManejoRemediacion='isChecked'
V
manejoRemediacion ← 'SI'
F
CheckBoxAgroecologia='isChecked'
V
agroecologia ← 'SI'
F
CheckBoxNutricionVegetal='isChecked'
V
nutricionVegetal ← 'SI'
F
CheckBoxOtros='isChecked'
V
otros ← 'SI'
actividadotros ← editTextOtros
AsyncTask()
FinProceso

GRAFO


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
32
33
34
35
36
37
38
39
40
41
42

RUTAS

R1: 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, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R2: 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, 32, 33, 34, 35, 36, 37, 38, 39, 41, 42
R3: 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, 32, 33, 34, 35, 36, 37, 39, 40, 41, 42
R4: 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, 32, 33, 34, 35, 37, 38, 39, 40, 41, 42
R5: 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, 32, 33, 35, 36, 37, 38, 39, 40, 41, 42
R6: 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, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R7: 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, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R8: 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, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R9: 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, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R10: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R11: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R12: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R13: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
R14: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 42

COMPLEJIDAD CICLOMÁTICA

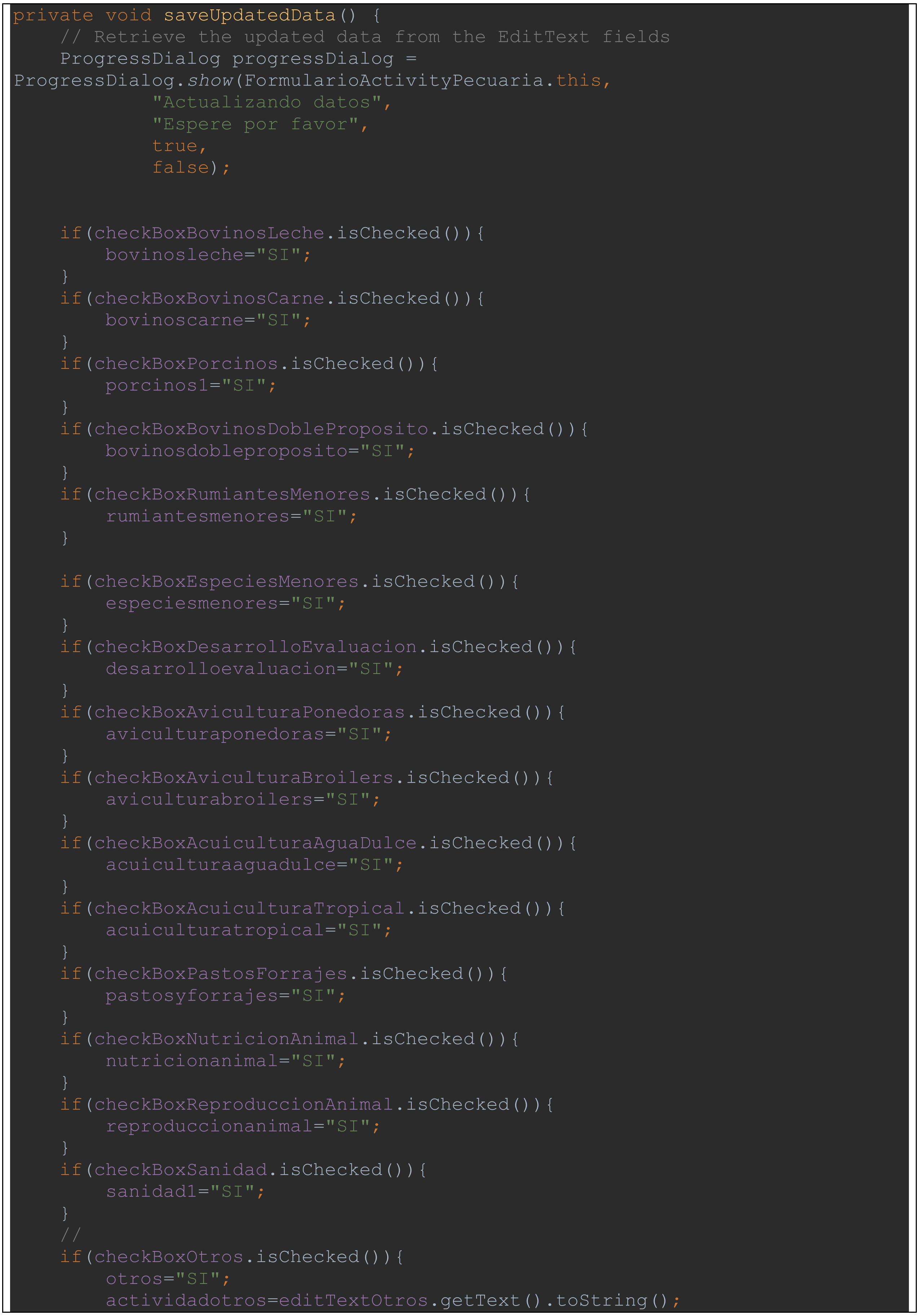
Se puede calcular de las siguientes formas:

- V(G) = número de nodos predicados(decisiones)+1
  V(G)=12+1=13
- V(G) = A – N + 2
  V(G)= 38– 27 + 2 = 13

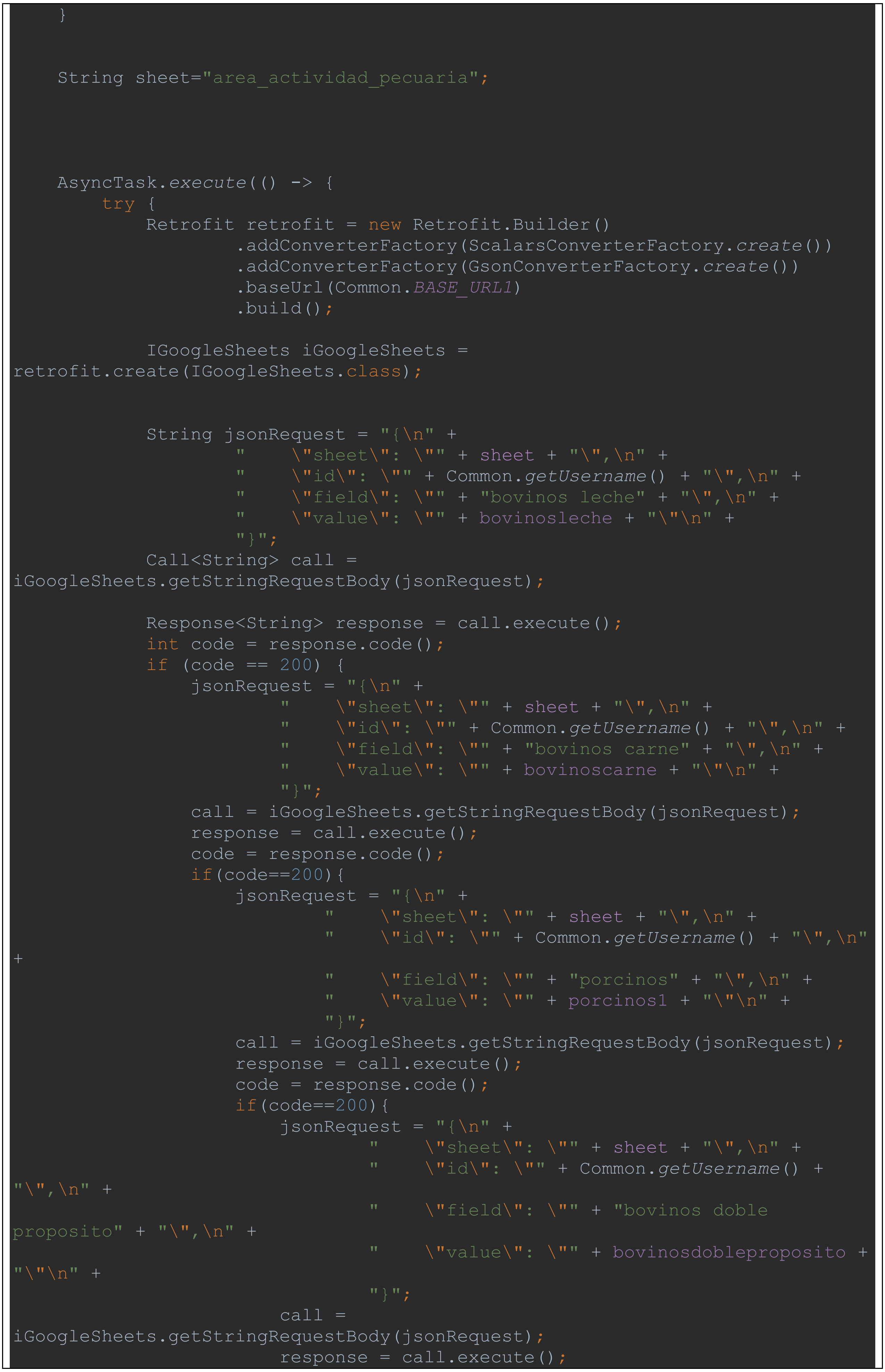
DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

Req. 12 ACTUALIZAR ACTIVIDAD PECUARIA

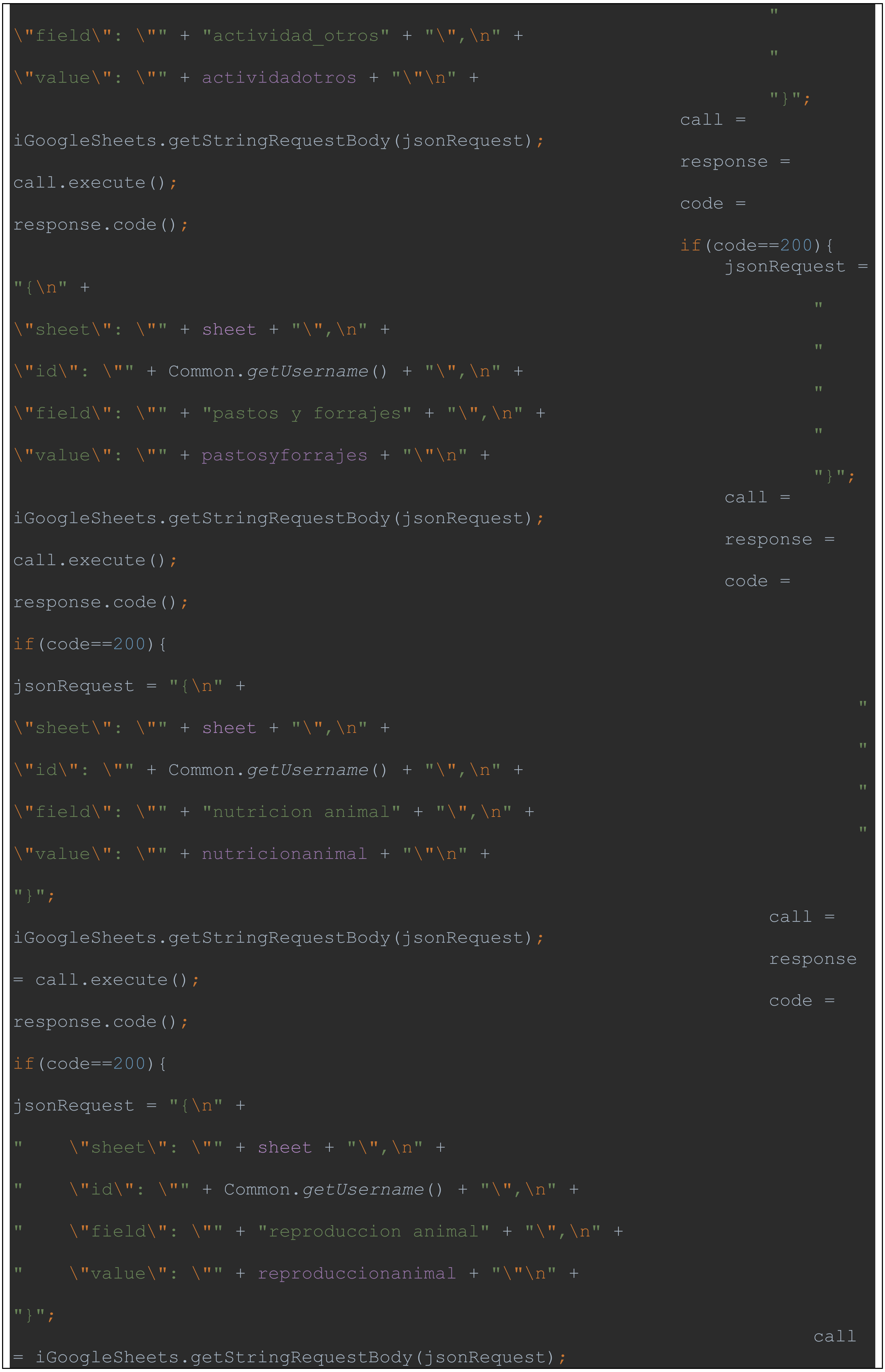
CÓDIGO FUENTE



```
private void saveUpdatedData() {
    // Retrieve the updated data from the EditText fields
    ProgressDialog progressDialog =
ProgressDialog.show(FormularioActivityPecuaria.this,
            "Actualizando datos",
            "Espere por favor",
            true,
            false);


    if(checkBoxBovinosLeche.isChecked()){
        bovinosleche="SI";
    }
    if(checkBoxBovinosCarne.isChecked()){
        bovinoscarne="SI";
    }
    if(checkBoxPorcinos.isChecked()){
        porcinos1="SI";
    }
    if(checkBoxBovinosDobleProposito.isChecked()){
        bovinosdobleproposito="SI";
    }
    if(checkBoxRumiantesMenores.isChecked()){
        rumiantesmenores="SI";
    }

    if(checkBoxEspeciesMenores.isChecked()){
        especiesmenores="SI";
    }
    if(checkBoxDesarrolloEvaluacion.isChecked()){
        desarrolloevaluacion="SI";
    }
    if(checkBoxAviculturaPonedoras.isChecked()){
        aviculturaponedoras="SI";
    }
    if(checkBoxAviculturaBroilers.isChecked()){
        aviculturabroilers="SI";
    }
    if(checkBoxAcuiculturaAguaDulce.isChecked()){
        acuiculturaaguadulce="SI";
    }
    if(checkBoxAcuiculturaTropical.isChecked()){
        acuiculturatropical="SI";
    }
    if(checkBoxPastosForrajes.isChecked()){
        pastosyforrajes="SI";
    }
    if(checkBoxNutricionAnimal.isChecked()){
        nutricionanimal="SI";
    }
    if(checkBoxReproduccionAnimal.isChecked()){
        reproduccionanimal="SI";
    }
    if(checkBoxSanidad.isChecked()){
        sanidad1="SI";
    }
    //
    if(checkBoxOtros.isChecked()){
        otros="SI";
        actividadotros=editTextOtros.getText().toString();
```

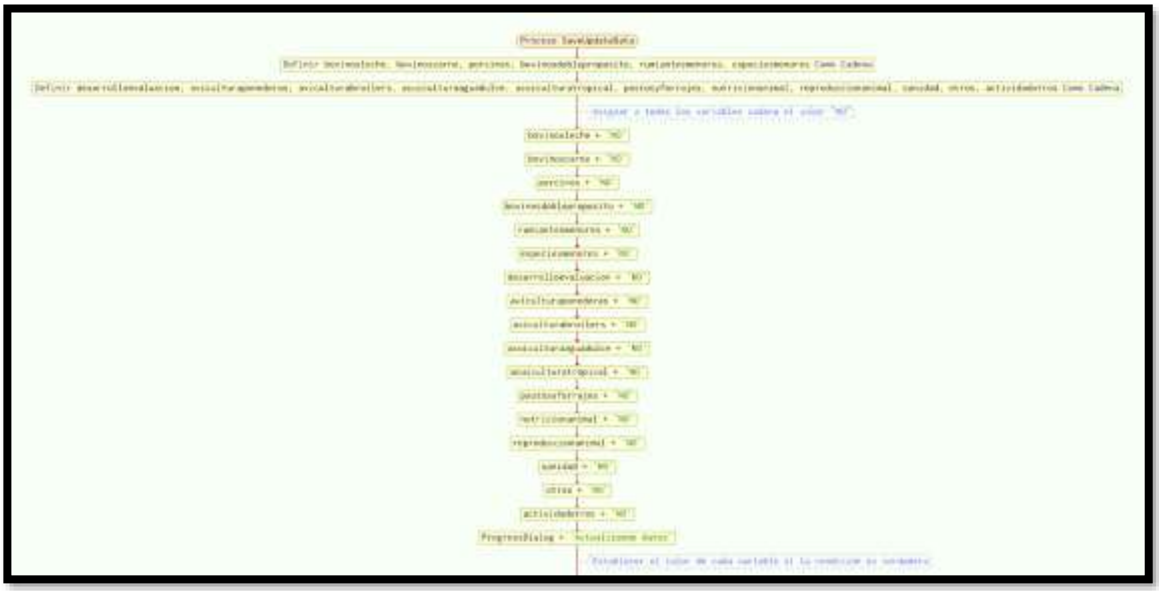


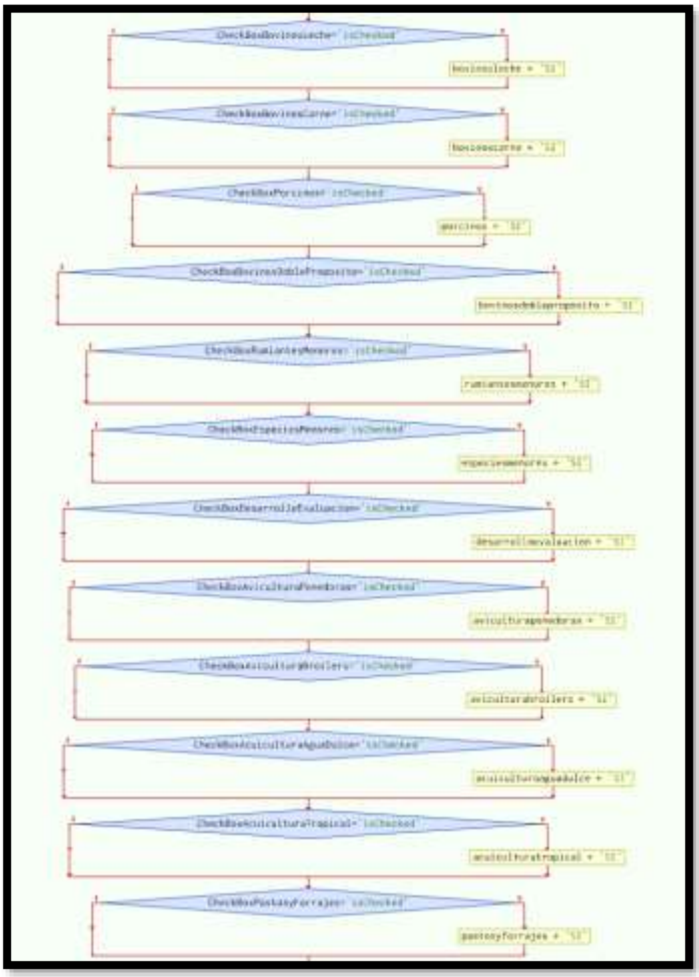
```
    }


    String sheet="area_actividad_pecuaria";




    AsyncTask.execute(() -> {
        try {
            Retrofit retrofit = new Retrofit.Builder()
                    .addConverterFactory(ScalarsConverterFactory.create())
                    .addConverterFactory(GsonConverterFactory.create())
                    .baseUrl(Common.BASE_URL1)
                    .build();

            IGoogleSheets iGoogleSheets =
retrofit.create(IGoogleSheets.class);


            String jsonRequest = "{\n" +
                    "    \"sheet\": \"" + sheet + "\",\n" +
                    "    \"id\": \"" + Common.getUsername() + "\",\n" +
                    "    \"field\": \"" + "bovinos leche" + "\",\n" +
                    "    \"value\": \"" + bovinosleche + "\"\n" +
                    "}";
            Call<String> call =
iGoogleSheets.getStringRequestBody(jsonRequest);

            Response<String> response = call.execute();
            int code = response.code();
            if (code == 200) {
                jsonRequest = "{\n" +
                        "    \"sheet\": \"" + sheet + "\",\n" +
                        "    \"id\": \"" + Common.getUsername() + "\",\n" +
                        "    \"field\": \"" + "bovinos carne" + "\",\n" +
                        "    \"value\": \"" + bovinoscarne + "\"\n" +
                        "}";
                call = iGoogleSheets.getStringRequestBody(jsonRequest);
                response = call.execute();
                code = response.code();
                if(code==200){
                    jsonRequest = "{\n" +
                            "    \"sheet\": \"" + sheet + "\",\n" +
                            "    \"id\": \"" + Common.getUsername() + "\",\n"
+
                            "    \"field\": \"" + "porcinos" + "\",\n" +
                            "    \"value\": \"" + porcinos1 + "\"\n" +
                            "}";
                    call = iGoogleSheets.getStringRequestBody(jsonRequest);
                    response = call.execute();
                    code = response.code();
                    if(code==200){
                        jsonRequest = "{\n" +
                                "    \"sheet\": \"" + sheet + "\",\n" +
                                "    \"id\": \"" + Common.getUsername() +
"\",\n" +
                                "    \"field\": \"" + "bovinos doble
proposito" + "\",\n" +
                                "    \"value\": \"" + bovinosdobleproposito +
"\"\n" +
                                "}";
                        call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                        response = call.execute();
                        code = response.code();
                        if(code==200){
                            jsonRequest = "{\n" +
                                    "    \"sheet\": \"" + sheet + "\",\n" +
                                    "    \"id\": \"" + Common.getUsername() +
"\",\n" +
                                    "    \"field\": \"" + "rumiantes menores"
+ "\",\n" +
                                    "    \"value\": \"" + rumiantesmenores +
"\"\n" +
                                    "}";
                            call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                            response = call.execute();
                            code = response.code();
                            if(code==200){
                                jsonRequest = "{\n" +
                                        "    \"sheet\": \"" + sheet + "\",\n"
+
                                        "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                        "    \"field\": \"" + "especies
menores" + "\",\n" +
                                        "    \"value\": \"" +
especiesmenores+ "\"\n" +
                                        "}";
                                call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                response = call.execute();
                                code = response.code();
                                if(code==200){
                                    jsonRequest = "{\n" +
                                            "    \"sheet\": \"" + sheet +
"\",\n" +
                                            "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                            "    \"field\": \"" + "desarrollo
y evaluacion de proyectos" + "\",\n" +
                                            "    \"value\": \"" +
desarrolloevaluacion + "\"\n" +
                                            "}";
                                    call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                    response = call.execute();
                                    code = response.code();
                                    if(code==200){
                                        jsonRequest = "{\n" +
                                                "    \"sheet\": \"" + sheet +
"\",\n" +
                                                "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                                "    \"field\": \"" +
"avicultura ponedoras" + "\",\n" +
                                                "    \"value\": \"" +
aviculturaponedoras + "\"\n" +
                                                "}";
                                        call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                        response = call.execute();
                                        code = response.code();
                                        if(code==200){
                                            jsonRequest = "{\n" +
                                                    "    \"sheet\": \"" +
sheet + "\",\n" +
                                                    "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                                    "    \"field\": \"" +
"avicultura broilers" + "\",\n" +
                                                    "    \"value\": \"" +
aviculturabroilers + "\"\n" +
                                                    "}";
                                            call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                            response = call.execute();
                                            code = response.code();
                                            if(code==200){
                                                jsonRequest = "{\n" +
                                                        "    \"sheet\": \"" +
sheet + "\",\n" +
                                                        "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                                        "    \"field\": \"" +
"acuicultura agua dulce" + "\",\n" +
                                                        "    \"value\": \"" +
acuiculturaaguadulce+ "\"\n" +
                                                        "}";
                                                call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                                response = call.execute();
                                                code = response.code();
                                                if(code==200){
                                                    jsonRequest = "{\n" +
                                                            "    \"sheet\":
\"" + sheet + "\",\n" +
                                                            "    \"id\": \""
+ Common.getUsername() + "\",\n" +
                                                            "    \"field\":
\"" + "acuicultura tropical" + "\",\n" +
                                                            "    \"value\":
\"" + acuiculturatropical + "\"\n" +
                                                            "}";
                                                    call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                                    response =
call.execute();
                                                    code = response.code();
                                                    if(code==200){
                                                        jsonRequest = "{\n" +
                                                                "
\"sheet\": \"" + sheet + "\",\n" +
                                                                "    \"id\":
\"" + Common.getUsername() + "\",\n" +
                                                                "
\"field\": \"" + "otros" + "\",\n" +
                                                                "
\"value\": \"" + otros + "\"\n" +
                                                                "}";
                                                        call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                                        response =
call.execute();
                                                        code =
response.code();
                                                        if(code==200){
                                                            jsonRequest =
"{\n" +
                                                                    "
\"sheet\": \"" + sheet + "\",\n" +
                                                                    "
\"id\": \"" + Common.getUsername() + "\",\n" +
```

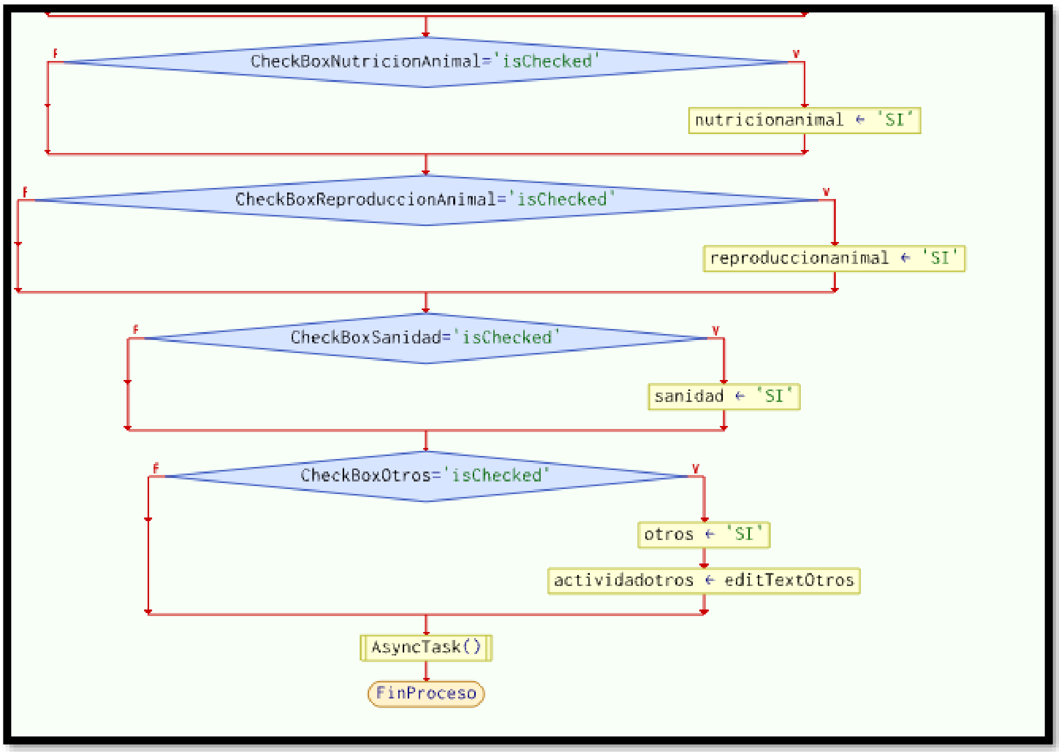


```
                                                                              "
\"field\": \"" + "actividad_otros" + "\",\n" +
                                                                              "
\"value\": \"" + actividadotros + "\"\n" +
                                                                              "}";
                                                                      call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                                                      response =
call.execute();
                                                                      code =
response.code();
                                                                      if(code==200){
                                                                          jsonRequest =
"{\n" +
                                                                                  "
\"sheet\": \"" + sheet + "\",\n" +
                                                                                  "
\"id\": \"" + Common.getUsername() + "\",\n" +
                                                                                  "
\"field\": \"" + "pastos y forrajes" + "\",\n" +
                                                                                  "
\"value\": \"" + pastosyforrajes + "\"\n" +
                                                                                  "}";
                                                                          call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                                                          response =
call.execute();
                                                                          code =
response.code();
if(code==200){
jsonRequest = "{\n" +
                                                                                      "
\"sheet\": \"" + sheet + "\",\n" +
                                                                                      "
\"id\": \"" + Common.getUsername() + "\",\n" +
                                                                                      "
\"field\": \"" + "nutricion animal" + "\",\n" +
                                                                                      "
\"value\": \"" + nutricionanimal + "\"\n" +
"}";
                                                                              call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                                                              response
= call.execute();
                                                                              code =
response.code();
if(code==200){
jsonRequest = "{\n" +
"     \"sheet\": \"" + sheet + "\",\n" +
"     \"id\": \"" + Common.getUsername() + "\",\n" +
"     \"field\": \"" + "reproduccion animal" + "\",\n" +
"     \"value\": \"" + reproduccionanimal + "\"\n" +
"}";
                                                                                  call
= iGoogleSheets.getStringRequestBody(jsonRequest);
response = call.execute();
                                                                        code
= response.code();

if(code==200){

jsonRequest = "{\n" +

"    \"sheet\": \"" + sheet + "\",\n" +

"    \"id\": \"" + Common.getUsername() + "\",\n" +

"    \"field\": \"" + "sanidad" + "\",\n" +

"    \"value\": \"" + sanidad1 + "\"\n" +

"}";

call = iGoogleSheets.getStringRequestBody(jsonRequest);

response = call.execute();

code = response.code();

if(code==200){

return;
                                                                            }
                                                                        }
                                                                    }
                                                                }
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                    progressDialog.dismiss();
                }

            }


        } catch (Exception e) {
            e.printStackTrace();
        }
    });
}
```

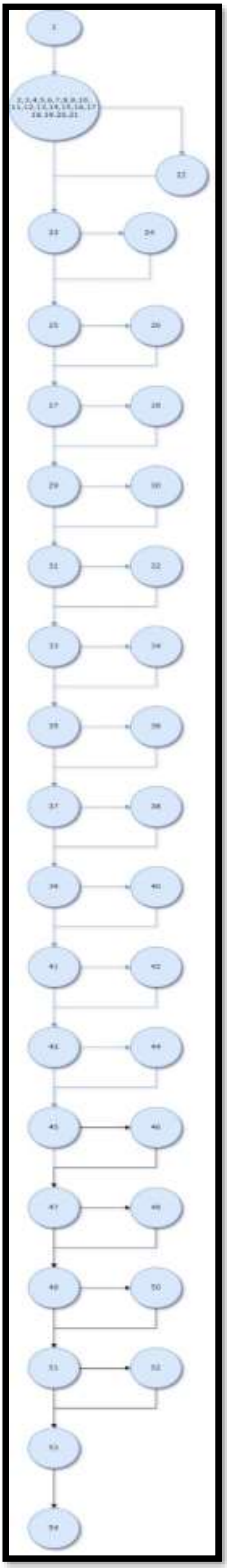
DIAGRAMA DE FLUJO


Proceso SaveUpdateData
Definir bovinosleche, bovinoscarne, porcinos, bovinosdobleproposito, rumiantesmenores, especiesmenores Como Cadena
Definir desarrolloevaluacion, aviculturaponedoras, aviculturabroilers, acuiculturaaguadulce, acuiculturatropical, pastosyforrajes, nutricionanimal, reproduccionanimal, sanidad, otros, actividadotros Como Cadena
Asignar a todas las variables cadena el valor "NO"
bovinosleche ← 'NO'
bovinoscarne ← 'NO'
porcinos ← 'NO'
bovinosdobleproposito ← 'NO'
rumiantesmenores ← 'NO'
especiesmenores ← 'NO'
desarrolloevaluacion ← 'NO'
aviculturaponedoras ← 'NO'
aviculturabroilers ← 'NO'
acuiculturaaguadulce ← 'NO'
acuiculturatropical ← 'NO'
pastosyforrajes ← 'NO'
nutricionanimal ← 'NO'
reproduccionanimal ← 'NO'
sanidad ← 'NO'
otros ← 'NO'
actividadotros ← 'NO'
ProgressDialog ← 'Actualizando datos'
Establecer el valor de cada variable si la condicion es verdadera


CheckBoxBovinosLeche='isChecked'
bovinosleche ← 'SI'
CheckBoxBovinosCarne='isChecked'
bovinoscarne ← 'SI'
CheckBoxPorcinos='isChecked'
porcinos ← 'SI'
CheckBoxBovinosDobleProposito='isChecked'
bovinosdobleproposito ← 'SI'
CheckBoxRumiantesMenores='isChecked'
rumiantesmenores ← 'SI'
CheckBoxEspeciesMenores='isChecked'
especiesmenores ← 'SI'
CheckBoxDesarrolloEvaluacion='isChecked'
desarrolloevaluacion ← 'SI'
CheckBoxAviculturaPonedoras='isChecked'
aviculturaponedoras ← 'SI'
CheckBoxAviculturaBroilers='isChecked'
aviculturabroilers ← 'SI'
CheckBoxAcuiculturaAguaDulce='isChecked'
acuiculturaaguadulce ← 'SI'
CheckBoxAcuiculturaTropical='isChecked'
acuiculturatropical ← 'SI'
CheckBoxPastosyForrajes='isChecked'
pastosyforrajes ← 'SI'



CheckBoxNutricionAnimal='isChecked'
F
V
nutricionanimal ← 'SI'
CheckBoxReproduccionAnimal='isChecked'
F
V
reproduccionanimal ← 'SI'
CheckBoxSanidad='isChecked'
F
V
sanidad ← 'SI'
CheckBoxOtros='isChecked'
F
V
otros ← 'SI'
actividadotros ← editTextOtros
AsyncTask()
FinProceso


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
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

GRAFO

RUTAS

R1: 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, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54

R2: 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, 32, 33, 34, 35, 36, 37, 38, 39, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54

R3: 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, 32, 33, 34, 35, 36, 37, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54

R4: 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, 32, 33, 34, 35, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54

R5: 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, 32, 33, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54

R6: 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, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54

R7: 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, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54

R8: 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, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54

R9: 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, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54

R10: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54

R11: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54

R12: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54

R13: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54

R14: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- V(G) = número de nodos predicados(decisiones)+1
  V(G)=16+1=17
- V(G) = A – N + 2
  V(G)= 50– 35 + 2 = 17

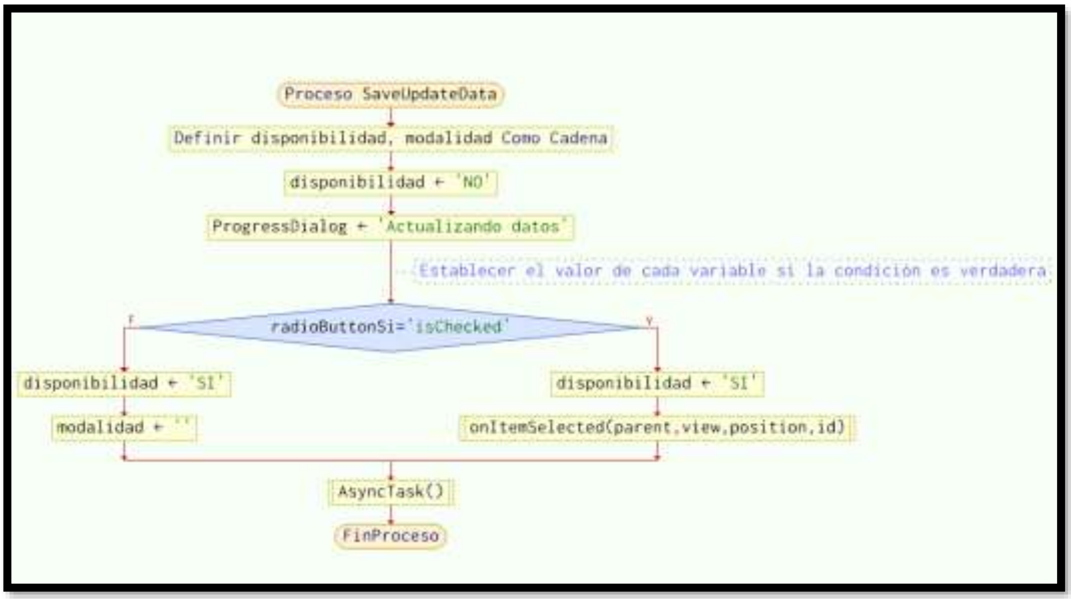
DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

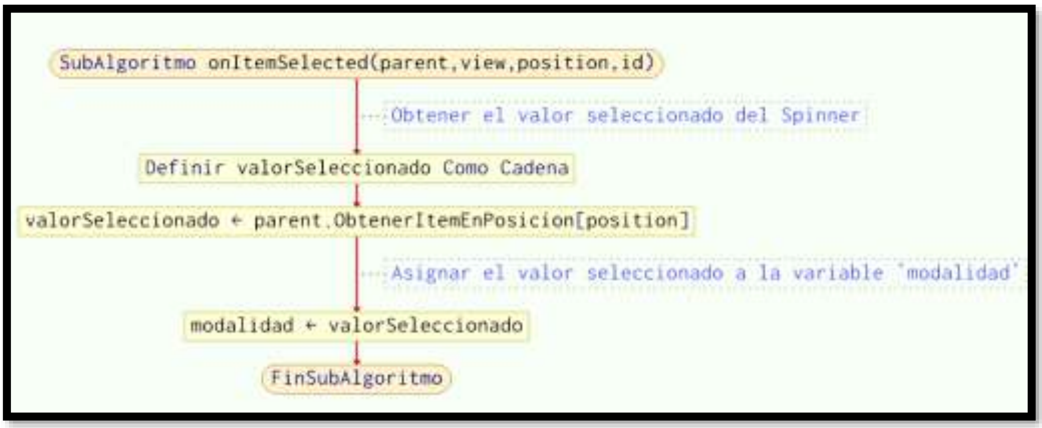
Req. 13 ACTUALIZAR DISPONIBILIDAD CONFERENCIAS

CÓDIGO FUENTE

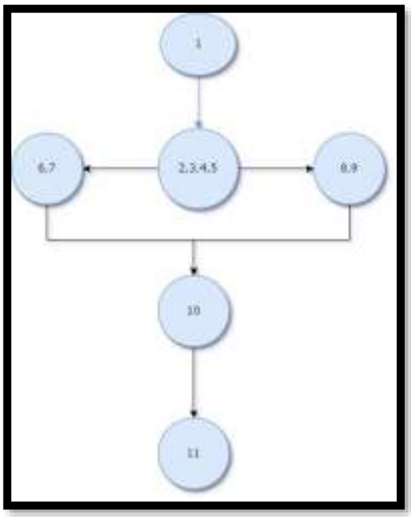
```
private void saveUpdatedData() {
    // Retrieve the updated data from the EditText fields
    ProgressDialog progressDialog =
ProgressDialog.show(FormularioConferencias.this,
            "Actualizando datos",
            "Espere por favor",
            true,
            false);
    if(radioButtonSi.isChecked()){
        disponibilidad="SI";
    }else{
        disponibilidad="NO";
        modalidad="";
    }
    String sheet="disponibilidad_conferencias";
    AsyncTask.execute(() -> {
        try {
            Retrofit retrofit = new Retrofit.Builder()
                    .addConverterFactory(ScalarsConverterFactory.create())
                    .addConverterFactory(GsonConverterFactory.create())
                    .baseUrl(Common.BASE_URL1)
                    .build();
            IGoogleSheets iGoogleSheets =
retrofit.create(IGoogleSheets.class);

            String jsonRequest = "{\n" +
                    "    \"sheet\": \"" + sheet + "\",\n" +
                    "    \"id\": \"" + Common.getUsername() + "\",\n" +
                    "    \"field\": \"" + "disponibilidad" + "\",\n" +
                    "    \"value\": \"" + disponibilidad + "\"\n" +
                    "}";
            Call<String> call =
iGoogleSheets.getStringRequestBody(jsonRequest);

            Response<String> response = call.execute();
            int code = response.code();
            if (code == 200) {
                jsonRequest = "{\n" +
                        "    \"sheet\": \"" + sheet + "\",\n" +
                        "    \"id\": \"" + Common.getUsername() + "\",\n" +
                        "    \"field\": \"" + "modalidad" + "\",\n" +
                        "    \"value\": \"" + modalidad + "\"\n" +
                        "}";
                call = iGoogleSheets.getStringRequestBody(jsonRequest);
                response = call.execute();
                code = response.code();
                if(code==200){
                    return;
                }
            }
            progressDialog.dismiss();


        } catch (Exception e) {
            e.printStackTrace();
        }
    });
}
```

DIAGRAMA DE FLUJO


Proceso SaveUpdateData
Definir disponibilidad, modalidad Como Cadena
disponibilidad ← 'NO'
ProgressDialog ← 'Actualizando datos'
Establecer el valor de cada variable si la condición es verdadera
F
radioButtonSi='isChecked'
V
disponibilidad ← 'SI'
modalidad ← ''
disponibilidad ← 'SI'
onItemSelected(parent,view,position,id)
AsyncTask()
FinProceso


SubAlgoritmo onItemSelected(parent,view,position,id)
Obtener el valor seleccionado del Spinner
Definir valorSeleccionado Como Cadena
valorSeleccionado ← parent.ObtenerItemEnPosicion[position]
Asignar el valor seleccionado a la variable 'modalidad'
modalidad ← valorSeleccionado
FinSubAlgoritmo

GRAFO



RUTAS

R1: 1, 2, 3, 4, 5, 6, 7, 10,11
R2: 1, 2, 3, 4, 5, 8,9, 10,11

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- V(G) = número de nodos predicados(decisiones)+1
  V(G)=1+1=2
- V(G) = A – N + 2
  V(G)= 6– 6 + 2 = 2

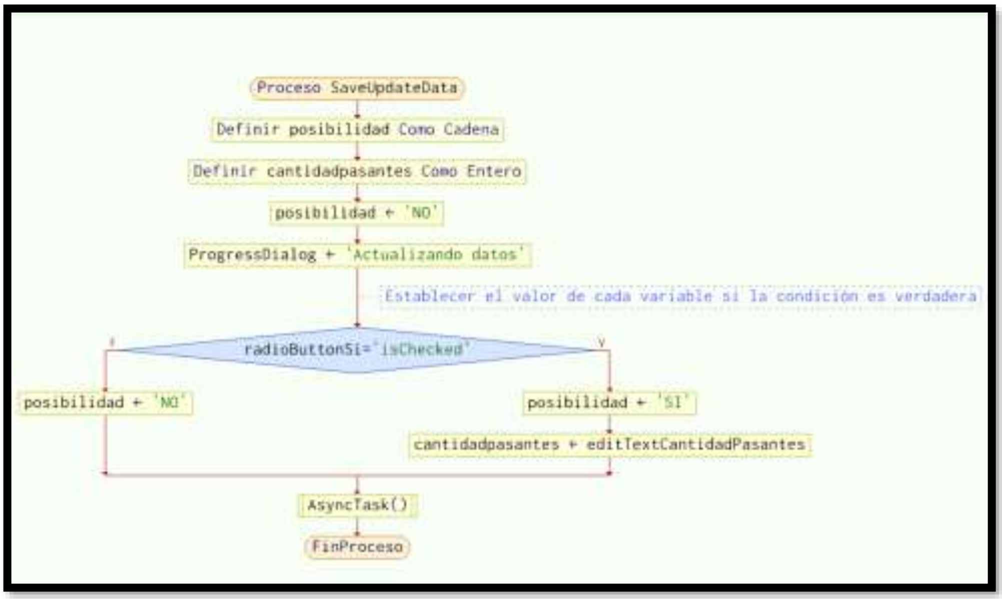
DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

Req. 14 ACTUALIZAR DISPONIBILIDAD PASANTIAS

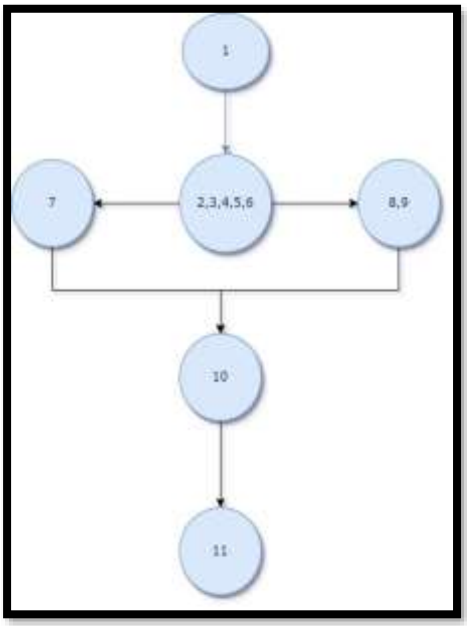
CÓDIGO FUENTE

```java
private void saveUpdatedData() {
    // Retrieve the updated data from the EditText fields
    ProgressDialog progressDialog =
ProgressDialog.show(FormularioPasantes.this,
            "Actualizando datos",
            "Espere por favor",
            true,
            false);
    if(radioButtonSi.isChecked()){
        posibilidad="SI";
        cantidadpasantes=editTextCantidadPasantes.getText().toString();
    }else{
        posibilidad="NO";
    }
    String sheet="posibilidad de pasantes";
    AsyncTask.execute(() -> {
        try {
            Retrofit retrofit = new Retrofit.Builder()
                    .addConverterFactory(ScalarsConverterFactory.create())
                    .addConverterFactory(GsonConverterFactory.create())
                    .baseUrl(Common.BASE_URL1)
                    .build();
            IGoogleSheets iGoogleSheets =
retrofit.create(IGoogleSheets.class);
            String jsonRequest = "{\n" +
                    "    \"sheet\": \"" + sheet + "\",\n" +
                    "    \"id\": \"" + Common.getUsername() + "\",\n" +
                    "    \"field\": \"" + "posibilidad" + "\",\n" +
                    "    \"value\": \"" + posibilidad + "\"\n" +
                    "}";
            Call<String> call =
iGoogleSheets.getStringRequestBody(jsonRequest);

            Response<String> response = call.execute();
            int code = response.code();
            if (code == 200) {
                jsonRequest = "{\n" +
                        "    \"sheet\": \"" + sheet + "\",\n" +
                        "    \"id\": \"" + Common.getUsername() + "\",\n" +
                        "    \"field\": \"" + "cantidad" + "\",\n" +
                        "    \"value\": \"" + cantidadpasantes + "\"\n" +
                        "}";
                call = iGoogleSheets.getStringRequestBody(jsonRequest);
                response = call.execute();
                code = response.code();
                if(code==200){
                    return;
                }
            }
            progressDialog.dismiss();

        } catch (Exception e) {
            e.printStackTrace();
        }
    });
}
```

DIAGRAMA DE FLUJO


Proceso SaveUpdateData
Definir posibilidad Como Cadena
Definir cantidadpasantes Como Entero
posibilidad ← 'NO'
ProgressDialog ← 'Actualizando datos'
Establecer el valor de cada variable si la condición es verdadera
radioButtonSi='isChecked'
F
V
posibilidad ← 'NO'
posibilidad ← 'SI'
cantidadpasantes ← editTextCantidadPasantes
AsyncTask()
FinProceso

GRAFO



RUTAS

R1: 1, 2, 3, 4, 5, 6, 7, 10,11
R2: 1, 2, 3, 4, 5, 6, 8,9, 10,11

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

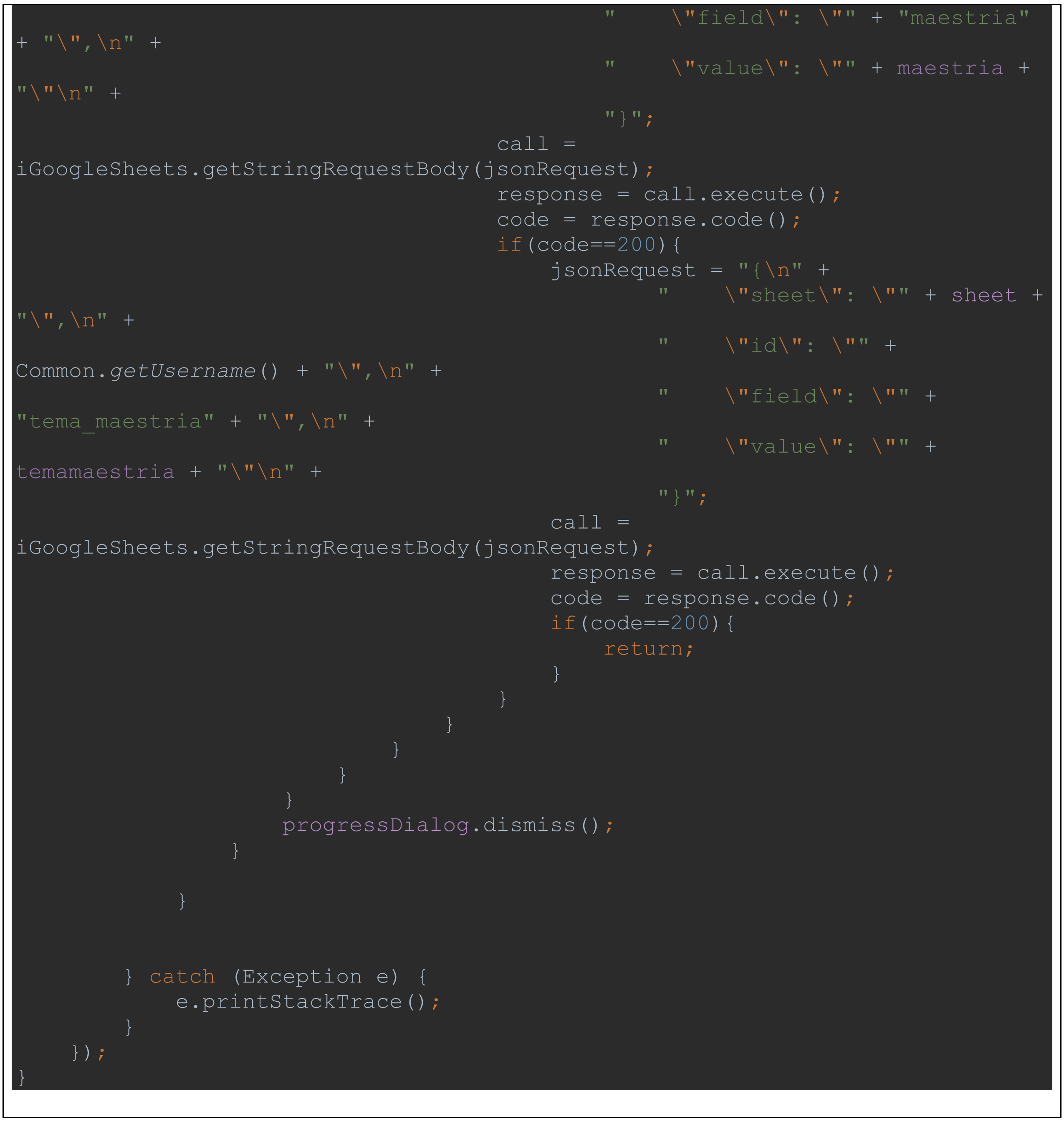
- V(G) = número de nodos predicados(decisiones)+1
  V(G)=1+1=2
- V(G) = A – N + 2
  V(G)= 6– 6 + 2 = 2

DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

Req. 15 ACTUALIZAR INFOMACIÓN CAPACITACIONES

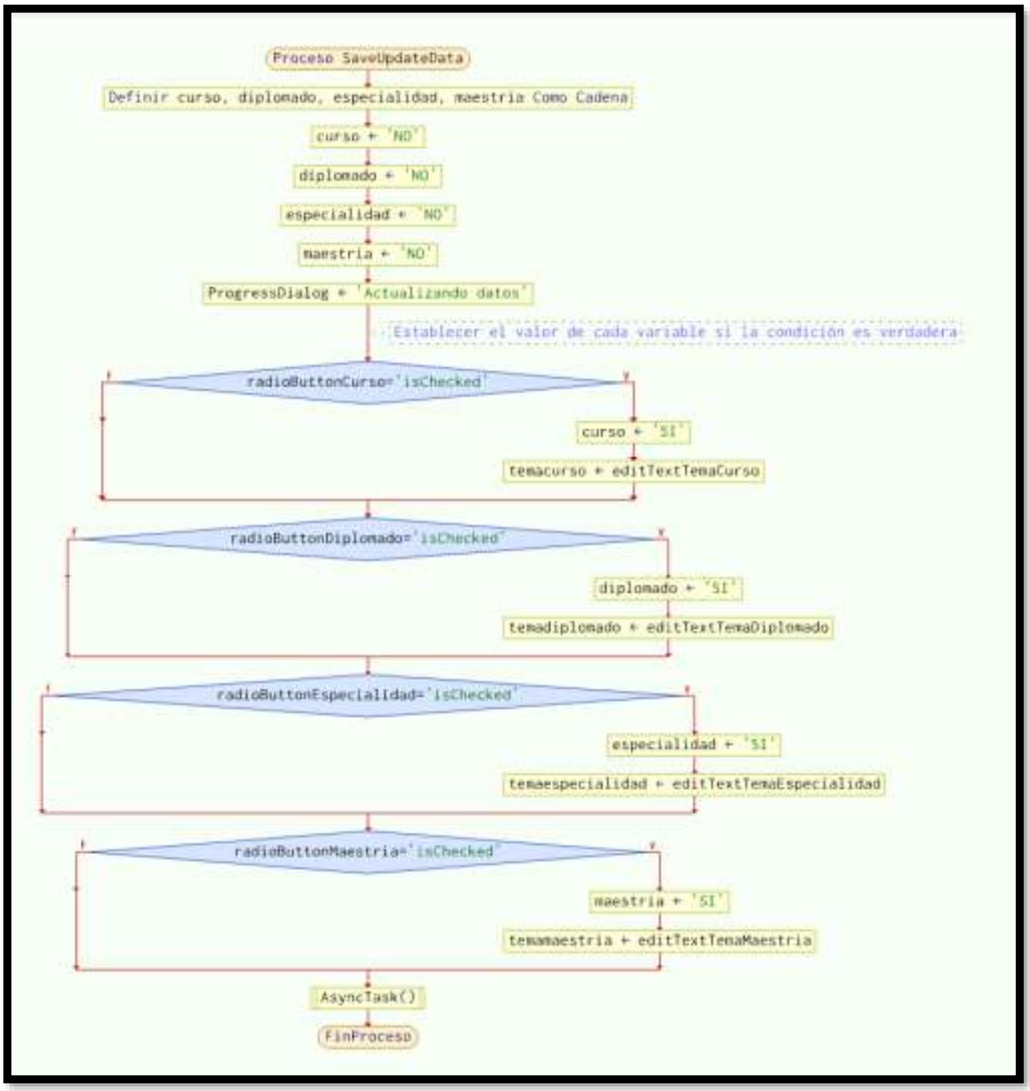
CÓDIGO FUENTE

```
private void saveUpdatedData() {
    // Retrieve the updated data from the EditText fields
    ProgressDialog progressDialog =
ProgressDialog.show(FormularioCapacitacion.this,
            "Actualizando datos",
            "Espere por favor",
            true,
            false);

    if(radioButtonSi.isChecked()){
        curso="SI";
        temacurso=editTextTemaCurso.getText().toString();
    }
    if(radioButtonSi1.isChecked()){
        diplomado="SI";
        temadiplomado=editTextTemaDiplomado.getText().toString();
    }
    if(radioButtonSi2.isChecked()){
        especialidad="SI";
        temaespecialidad=editTextTemaEspecialidad.getText().toString();
    }
    if(radioButtonSi3.isChecked()){
        maestria="SI";
        temamaestria=editTextTemaMaestria.getText().toString();
    }


    String sheet="necesidad de postgrados";




    AsyncTask.execute(() -> {
        try {
            Retrofit retrofit = new Retrofit.Builder()
                    .addConverterFactory(ScalarsConverterFactory.create())
                    .addConverterFactory(GsonConverterFactory.create())
                    .baseUrl(Common.BASE_URL1)
                    .build();

            IGoogleSheets iGoogleSheets =
retrofit.create(IGoogleSheets.class);


            String jsonRequest = "{\n" +
                    "    \"sheet\": \"" + sheet + "\",\n" +
                    "    \"id\": \"" + Common.getUsername() + "\",\n" +
                    "    \"field\": \"" + "curso" + "\",\n" +
                    "    \"value\": \"" + curso + "\"\n" +
                    "}";
            Call<String> call =
iGoogleSheets.getStringRequestBody(jsonRequest);

            Response<String> response = call.execute();
            int code = response.code();
            if (code == 200) {
                jsonRequest = "{\n" +
                        "    \"sheet\": \"" + sheet + "\",\n" +
                        "    \"id\": \"" + Common.getUsername() + "\",\n" +
                        "    \"field\": \"" + "tema_curso" + "\",\n" +
                        "    \"value\": \"" + temacurso + "\"\n" +
                        "}";
                call = iGoogleSheets.getStringRequestBody(jsonRequest);
                response = call.execute();
                code = response.code();
                if(code==200){
                    jsonRequest = "{\n" +
                            "    \"sheet\": \"" + sheet + "\",\n" +
                            "    \"id\": \"" + Common.getUsername() + "\",\n"
+
                            "    \"field\": \"" + "diplomado" + "\",\n" +
                            "    \"value\": \"" + diplomado + "\"\n" +
                            "}";
                    call = iGoogleSheets.getStringRequestBody(jsonRequest);
                    response = call.execute();
                    code = response.code();
                    if(code==200){
                        jsonRequest = "{\n" +
                                "    \"sheet\": \"" + sheet + "\",\n" +
                                "    \"id\": \"" + Common.getUsername() +
"\",\n" +
                                "    \"field\": \"" + "tema_diplomado" +
"\",\n" +
                                "    \"value\": \"" + temadiplomado + "\"\n"
+
                                "}";
                        call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                        response = call.execute();
                        code = response.code();
                        if(code==200){
                            jsonRequest = "{\n" +
                                    "    \"sheet\": \"" + sheet + "\",\n" +
                                    "    \"id\": \"" + Common.getUsername() +
"\",\n" +
                                    "    \"field\": \"" + "especialidad" +
"\",\n" +
                                    "    \"value\": \"" + especialidad +
"\"\n" +
                                    "}";
                            call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                            response = call.execute();
                            code = response.code();
                            if(code==200){
                                jsonRequest = "{\n" +
                                        "    \"sheet\": \"" + sheet + "\",\n"
+
                                        "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                        "    \"field\": \"" +
"tema_especialidad" + "\",\n" +
                                        "    \"value\": \"" +
temaespecialidad + "\"\n" +
                                        "}";
                                call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                response = call.execute();
                                code = response.code();
                                if(code==200){
                                    jsonRequest = "{\n" +
                                            "    \"sheet\": \"" + sheet +
"\",\n" +
                                            "    \"id\": \"" +
Common.getUsername() + "\",\n" +
```

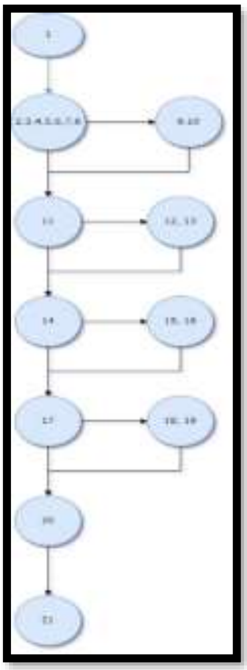


```
                                            "    \"field\": \"" + "maestria"
+ "\",\n" +
                                            "    \"value\": \"" + maestria +
"\"\n" +
                                            "}";
                                    call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                    response = call.execute();
                                    code = response.code();
                                    if(code==200){
                                        jsonRequest = "{\n" +
                                                "    \"sheet\": \"" + sheet +
"\",\n" +
                                                "    \"id\": \"" +
Common.getUsername() + "\",\n" +
                                                "    \"field\": \"" +
"tema_maestria" + "\",\n" +
                                                "    \"value\": \"" +
temamaestria + "\"\n" +
                                                "}";
                                        call =
iGoogleSheets.getStringRequestBody(jsonRequest);
                                        response = call.execute();
                                        code = response.code();
                                        if(code==200){
                                            return;
                                        }
                                    }
                                }
                            }
                        }
                    }
                    progressDialog.dismiss();
                }

            }


        } catch (Exception e) {
            e.printStackTrace();
        }
    });
}
```

DIAGRAMA DE FLUJO


Proceso SaveUpdateData
Definir curso, diplomado, especialidad, maestria Como Cadena
curso ← 'NO'
diplomado ← 'NO'
especialidad ← 'NO'
maestria ← 'NO'
ProgressDialog ← 'Actualizando datos'
Establecer el valor de cada variable si la condición es verdadera
radioButtonCurso='isChecked'
f
v
curso ← 'SI'
temacurso ← editTextTemaCurso
radioButtonDiplomado='isChecked'
f
v
diplomado ← 'SI'
temadiplomado ← editTextTemaDiplomado
radioButtonEspecialidad='isChecked'
f
v
especialidad ← 'SI'
temaespecialidad ← editTextTemaEspecialidad
radioButtonMaestria='isChecked'
f
v
maestria ← 'SI'
temamaestria ← editTextTemaMaestria
AsyncTask()
FinProceso

GRAFO



RUTAS

R1: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21
R2: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21
R3: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 17, 18, 19, 20, 21
R4: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, 15, 16, 17, 18, 19, 20, 21
R5: 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21
R6: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 17, 20, 21
R7: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, 15, 16, 17, 20, 21
R8: 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 20, 21
R9: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 17, 20, 21
R10: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, 17, 18, 19, 20, 21
R11: 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 17, 18, 19, 20, 21
R12: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, 15, 16, 17, 20, 21
R13: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, 17, 18, 19, 20, 21
R14: 1, 2, 3, 4, 5, 6, 7, 8, 11, 14, 15, 16, 17, 18, 19, 20, 21
R15: 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 20, 21
R16: 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 17, 18, 19, 20, 21
R17: 1, 2, 3, 4, 5, 6, 7, 8, 11, 14, 15, 16, 17, 18, 19, 20, 21
R18: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, 17, 20, 21
R19: 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 17, 20, 21
R20: 1, 2, 3, 4, 5, 6, 7, 8, 11, 14, 15, 16, 17, 20, 21
R21: 1, 2, 3, 4, 5, 6, 7, 8, 11, 14, 17, 18, 19, 20, 21
R22: 1, 2, 3, 4, 5, 6, 7, 8, 11, 14, 17, 20, 21

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- V(G) = número de nodos predicados(decisiones)+1

V(G)=4+1=5

- V(G) = A – N + 2

  V(G)= 14– 11+ 2 = 5

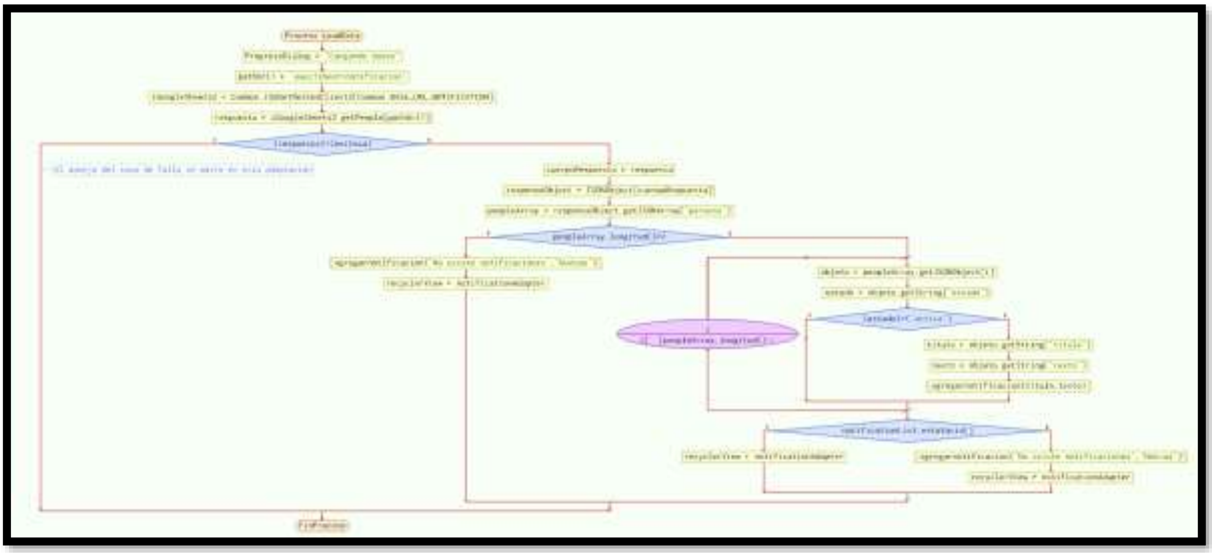
DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

Req. 16 VISUALIZAR NOTICIAS RELEVANTES

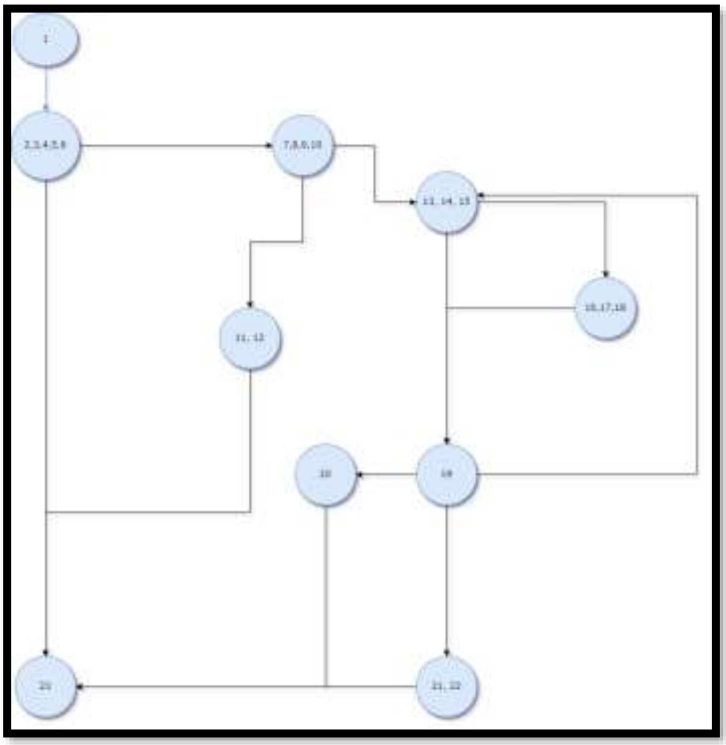
CÓDIGO FUENTE

```
public void LoadData(){
    ProgressDialog progressDialog = ProgressDialog.show(Notificaciones.this,
            "Cargando datos",
            "Espere por favor",
            true,
            false);
    iGoogleSheets2 =
Common.iGSGetMethodClient2(Common.BASE_URL_NOTIFICATION);
    String pathUrl1;
    pathUrl1 = "exec?sheet=notificacion";
    try {
        iGoogleSheets2.getPeople(pathUrl1).enqueue(new Callback<String>() {
            public void onResponse(@NonNull Call<String> call, @NonNull
Response<String> response) {
                try {
                    assert response.body() != null;
                    JSONObject responseObject = new
JSONObject(response.body());
                    JSONArray peopleArray =
responseObject.getJSONArray("persons");
                    JSONObject object;
                    if(peopleArray.length()>0){
                        for (int i = 0; i < peopleArray.length(); i++) {
                            object = peopleArray.getJSONObject(i);
                            String estado = object.getString("estado");
                            if(estado.equals("activa")){
                                String titulo = object.getString("titulo");
                                String texto = object.getString("texto");
                                notificationList.add(new Notification(titulo,
texto));

                            }
                        }
                        if(notificationList.isEmpty()){
                            notificationList.add(new Notification("No existe
notificaciones", "Nuevas"));
                            notificationAdapter = new
NotificationAdapter(notificationList);
                            recyclerView.setAdapter(notificationAdapter);

                        }
                        notificationAdapter = new
NotificationAdapter(notificationList);
                        recyclerView.setAdapter(notificationAdapter);
                    }else{
                        notificationList.add(new Notification("No existe
notificaciones", "Nuevas"));
                        notificationAdapter = new
NotificationAdapter(notificationList);
                        recyclerView.setAdapter(notificationAdapter);

                    }
                    progressDialog.dismiss();
                } catch (JSONException je) {
                    je.printStackTrace();
                }
            }

            @Override
            public void onFailure(@NonNull Call<String> call, @NonNull
Throwable t) {
            }
        });
    } catch (Exception e) {
        e.printStackTrace();
    }

}
```

DIAGRAMA DE FLUJO



GRAFO



RUTAS

R1: 1, 2,3,4,5,6,23
R2: 1,2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 11 , 12 , 13 , 14 , 15 , 16 , 17 , 18 , 19 , 20 , 21 , 22 , 23
R3: 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 11 , 12 , 13 , 14 , 15 , 16 , 17 , 18 , 19 , 20 , 21 , 22
R4: 1 , 2 , 3 , 4 , 5 , 23

COMPLEJIDAD CICLOMÁTICA

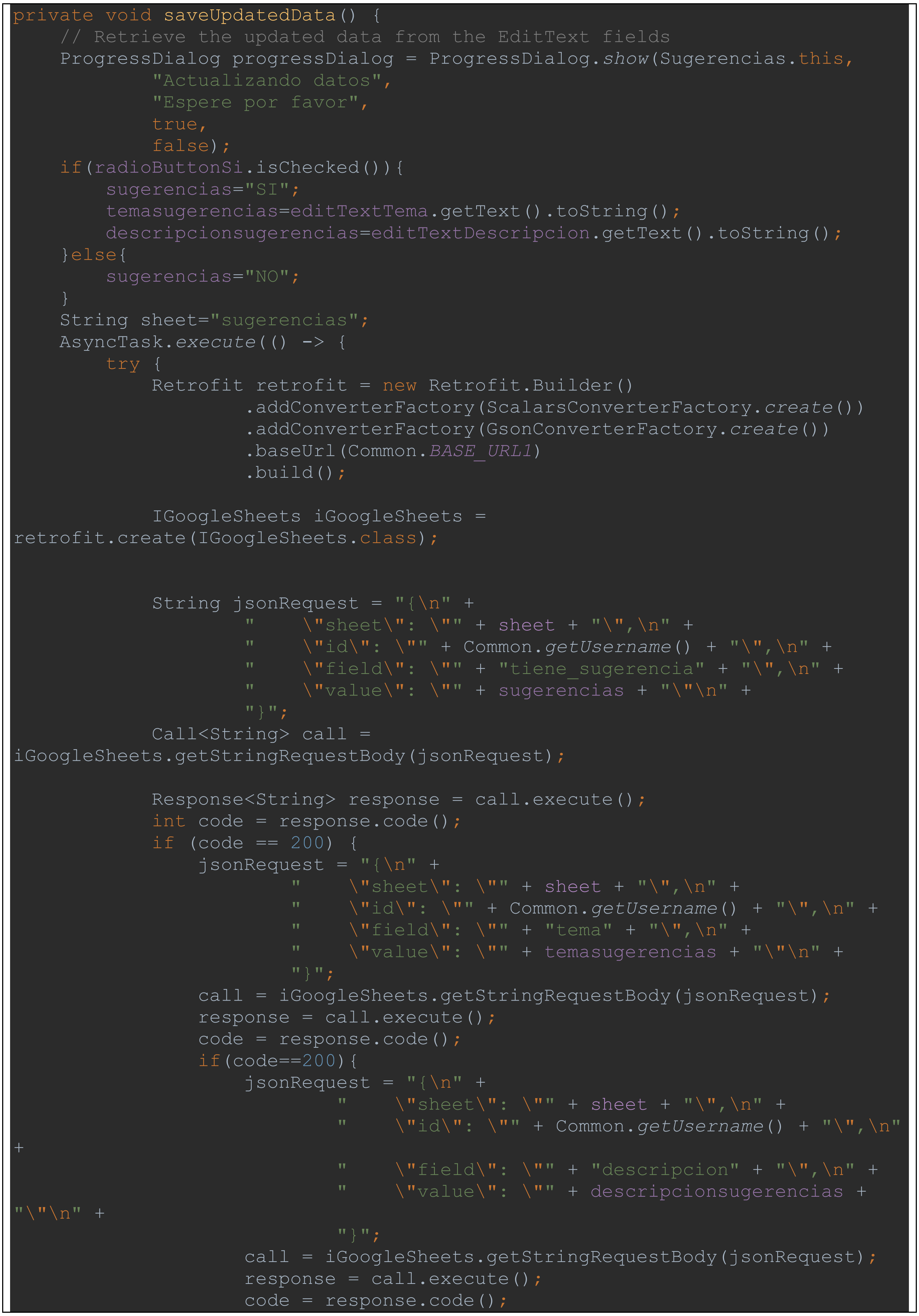
Se puede calcular de las siguientes formas:

- V(G) = número de nodos predicados(decisiones)+1
  V(G)=4+1=5
- V(G) = A – N + 2
  V(G)= 12– 9+ 2 = 5

DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos

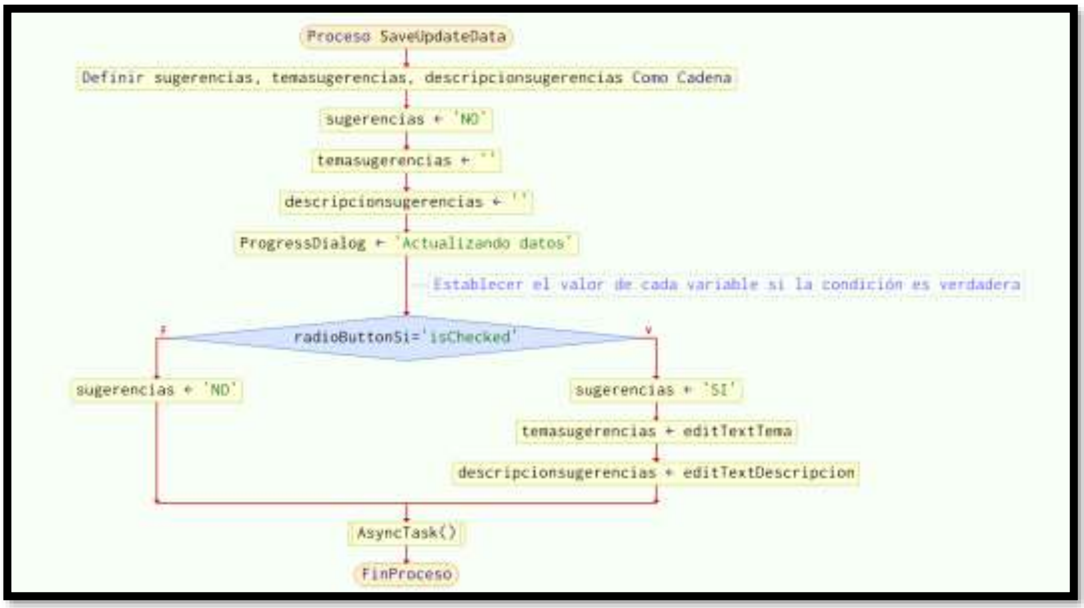
Req. 17 ENVIAR SUGERENCIAS

CÓDIGO FUENTE

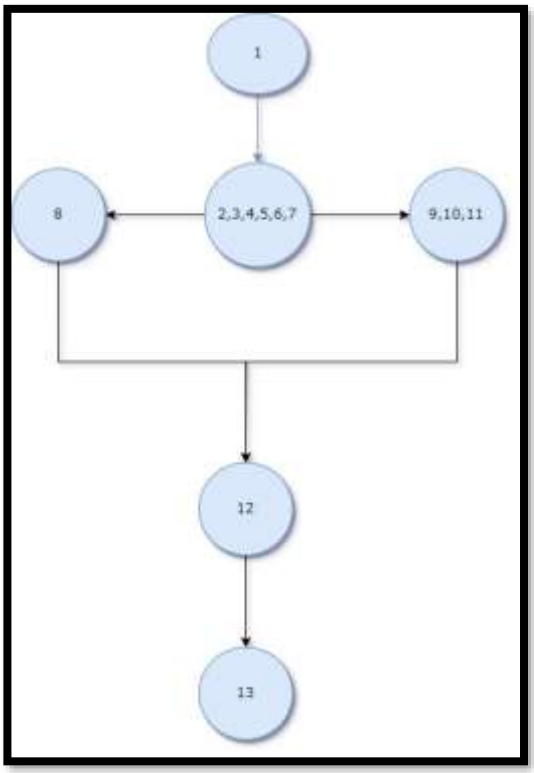


```java
private void saveUpdatedData() {
    // Retrieve the updated data from the EditText fields
    ProgressDialog progressDialog = ProgressDialog.show(Sugerencias.this,
            "Actualizando datos",
            "Espere por favor",
            true,
            false);
    if(radioButtonSi.isChecked()){
        sugerencias="SI";
        temasugerencias=editTextTema.getText().toString();
        descripcionsugerencias=editTextDescripcion.getText().toString();
    }else{
        sugerencias="NO";
    }
    String sheet="sugerencias";
    AsyncTask.execute(() -> {
        try {
            Retrofit retrofit = new Retrofit.Builder()
                    .addConverterFactory(ScalarsConverterFactory.create())
                    .addConverterFactory(GsonConverterFactory.create())
                    .baseUrl(Common.BASE_URL1)
                    .build();

            IGoogleSheets iGoogleSheets =
retrofit.create(IGoogleSheets.class);


            String jsonRequest = "{\n" +
                    "    \"sheet\": \"" + sheet + "\",\n" +
                    "    \"id\": \"" + Common.getUsername() + "\",\n" +
                    "    \"field\": \"" + "tiene_sugerencia" + "\",\n" +
                    "    \"value\": \"" + sugerencias + "\"\n" +
                    "}";
            Call<String> call =
iGoogleSheets.getStringRequestBody(jsonRequest);

            Response<String> response = call.execute();
            int code = response.code();
            if (code == 200) {
                jsonRequest = "{\n" +
                        "    \"sheet\": \"" + sheet + "\",\n" +
                        "    \"id\": \"" + Common.getUsername() + "\",\n" +
                        "    \"field\": \"" + "tema" + "\",\n" +
                        "    \"value\": \"" + temasugerencias + "\"\n" +
                        "}";
                call = iGoogleSheets.getStringRequestBody(jsonRequest);
                response = call.execute();
                code = response.code();
                if(code==200){
                    jsonRequest = "{\n" +
                            "    \"sheet\": \"" + sheet + "\",\n" +
                            "    \"id\": \"" + Common.getUsername() + "\",\n"
+
                            "    \"field\": \"" + "descripcion" + "\",\n" +
                            "    \"value\": \"" + descripcionsugerencias +
"\"\n" +
                            "}";
                    call = iGoogleSheets.getStringRequestBody(jsonRequest);
                    response = call.execute();
                    code = response.code();
                    if(code==200){
                        return;
                    }
                    progressDialog.dismiss();
                }

            }


        } catch (Exception e) {
            e.printStackTrace();
        }
    });
}
```

DIAGRAMA DE FLUJO


Proceso SaveUpdateData
Definir sugerencias, temasugerencias, descripcionsugerencias Como Cadena
sugerencias ← 'NO'
temasugerencias ← ''
descripcionsugerencias ← ''
ProgressDialog ← 'Actualizando datos'
Establecer el valor de cada variable si la condición es verdadera
F
radioButtonSi='isChecked'
V
sugerencias ← 'NO'
sugerencias ← 'SI'
temasugerencias ← editTextTema
descripcionsugerencias ← editTextDescripcion
AsyncTask()
FinProceso

GRAFO



RUTAS

R1: 1, 2, 3, 4, 5, 6, 7, 8, 12, 13
R2: 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- V(G) = número de nodos predicados(decisiones)+1
  V(G)=1+1=2
- V(G) = A – N + 2
  V(G)= 6– 6+ 2 = 2

DONDE:
P: Número de nodos predicado
A: Número de aristas
N: Número de nodos