Universidad Autónoma del Estado de México

Centro Universitario UAEM Atlacomulco

Alumno: Gary Elizalde Marcial

Licenciatura en Ingeniería en Computación

Tercer semestre

Unidad de aprendizaje: Bases de datos

Evaluación Extraordinaria

Docente: Laura Colin Rivas

Diciembre 2021, Atlacomulco, Estado de México

Sistema de control de inventarios en Abarrotes usando Firebase

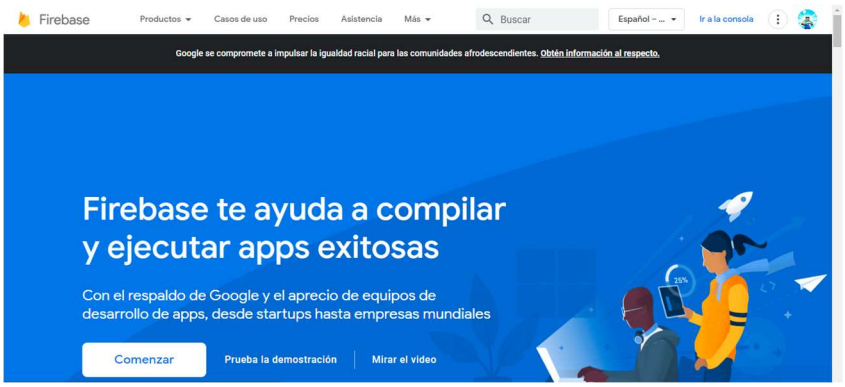
Para empezar, debemos de tener una cuenta de Google para poder usar su servicio de BD.

Navegaremos al siguiente sitio:

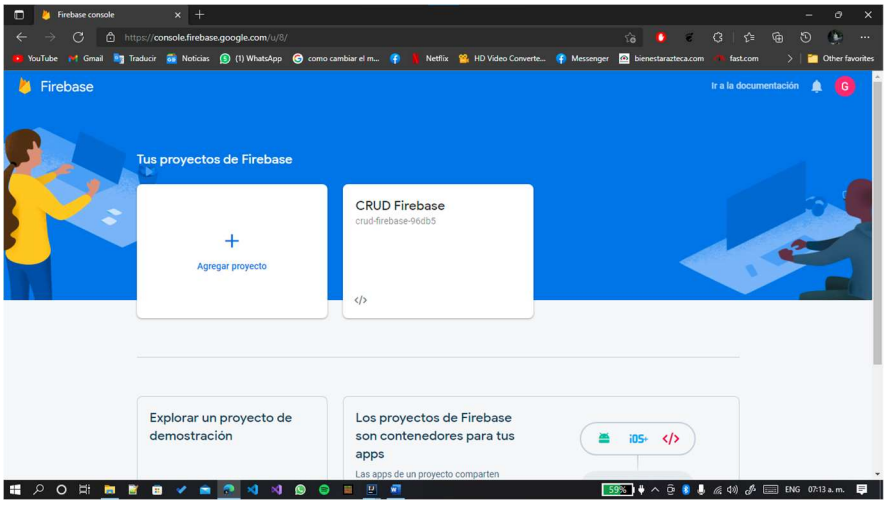
https://firebase.google.com

Nos mostrara la siguiente pantalla:

Daremos click en Empezar:

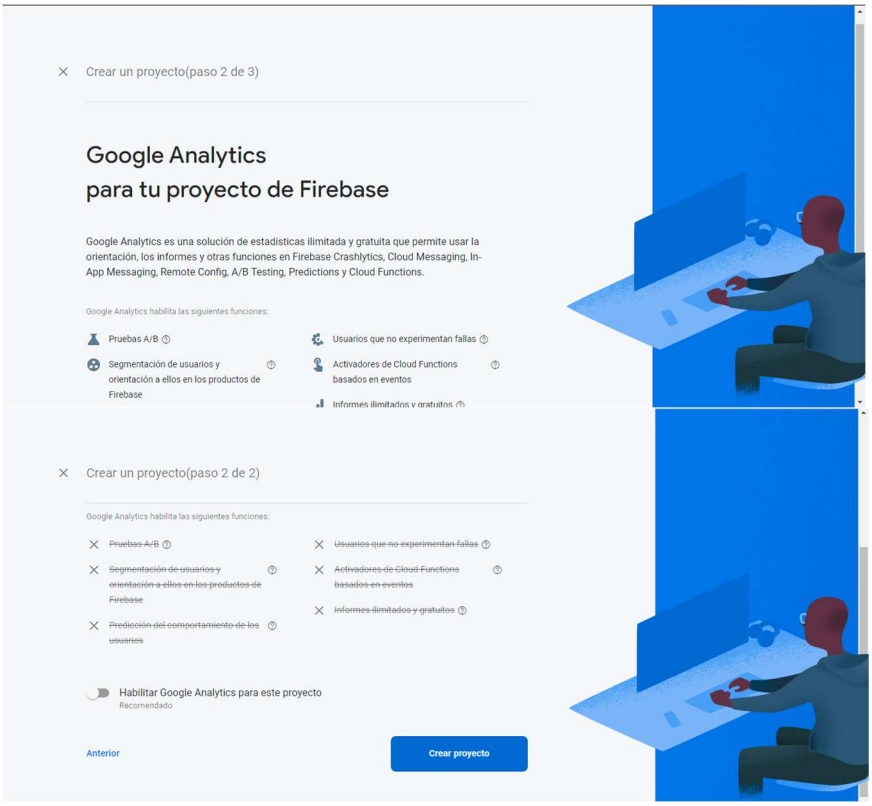


Hacemos click en “Agregar proyecto”

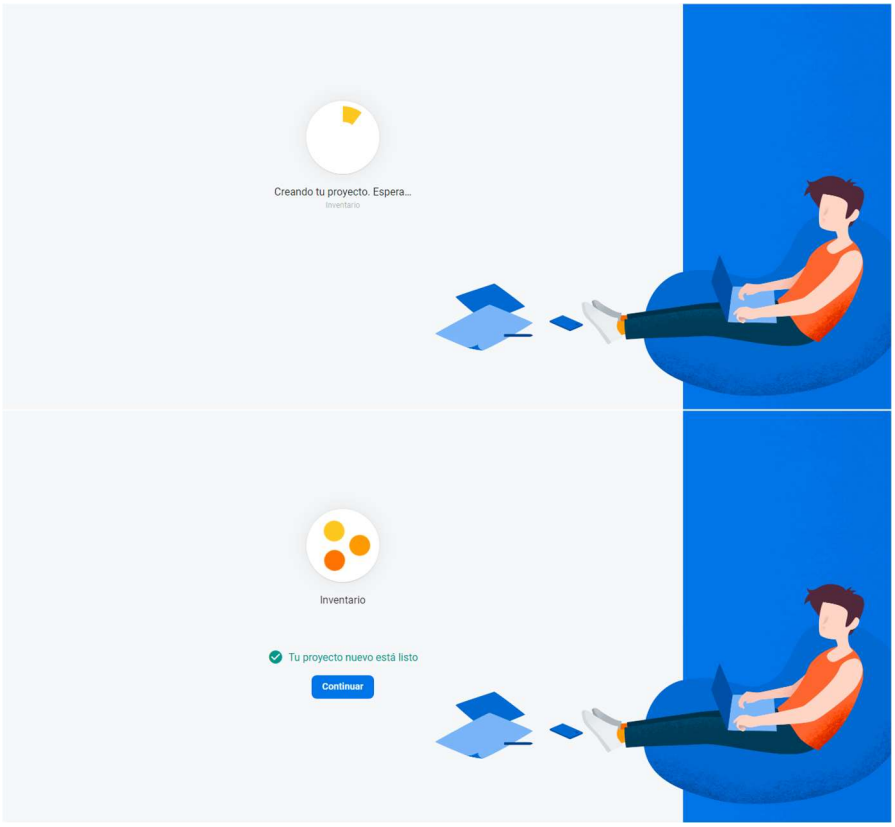


Y le damos un nombre:

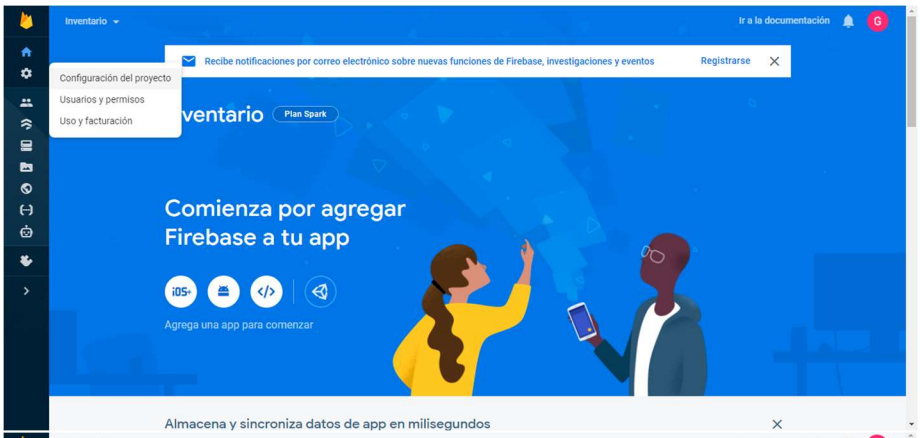
Desactivamos los Google analiticys ya que no necesitamos las métricas:

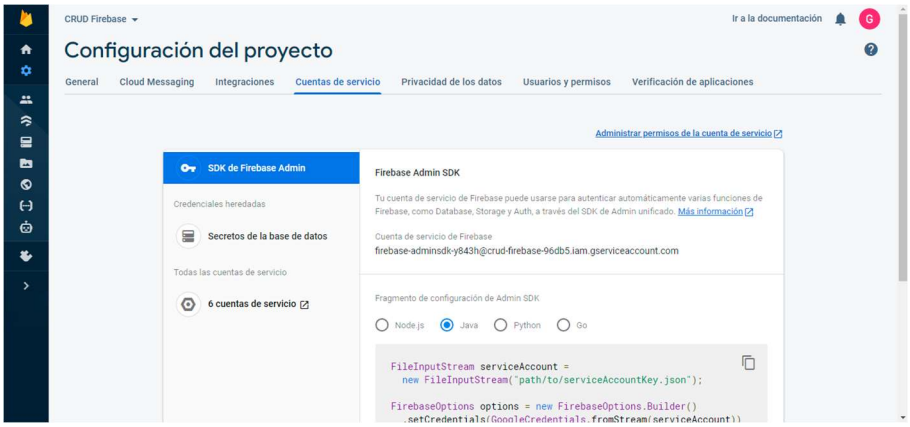


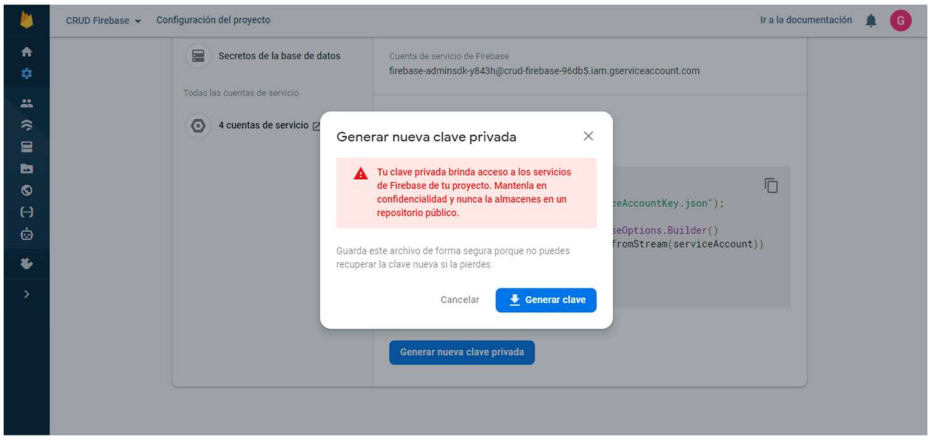
Una vez hecho esto empezara a crear el proyecto:



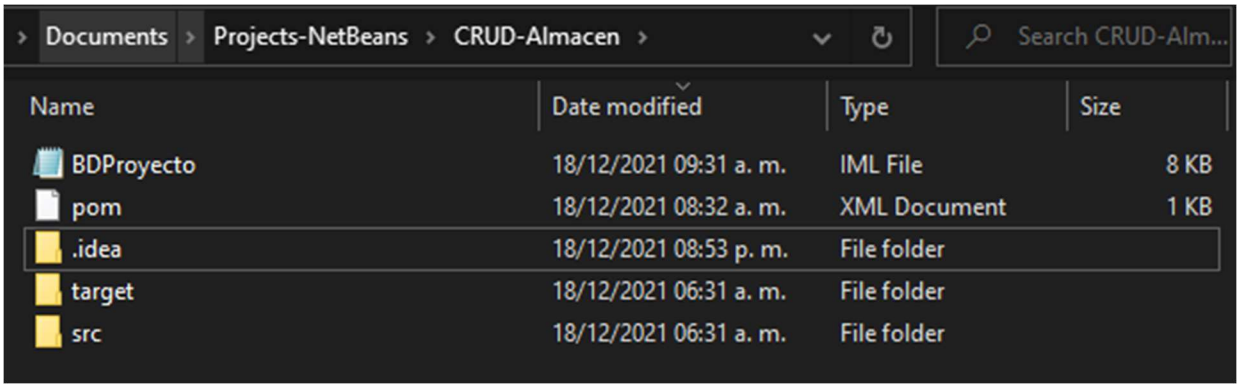
Nos dirigimos a la pestaña de configuración de proyecto y una vez dentro de esta exportaremos nuestra clave privada para poder conectar el proyecto de JAVA a Firebase:

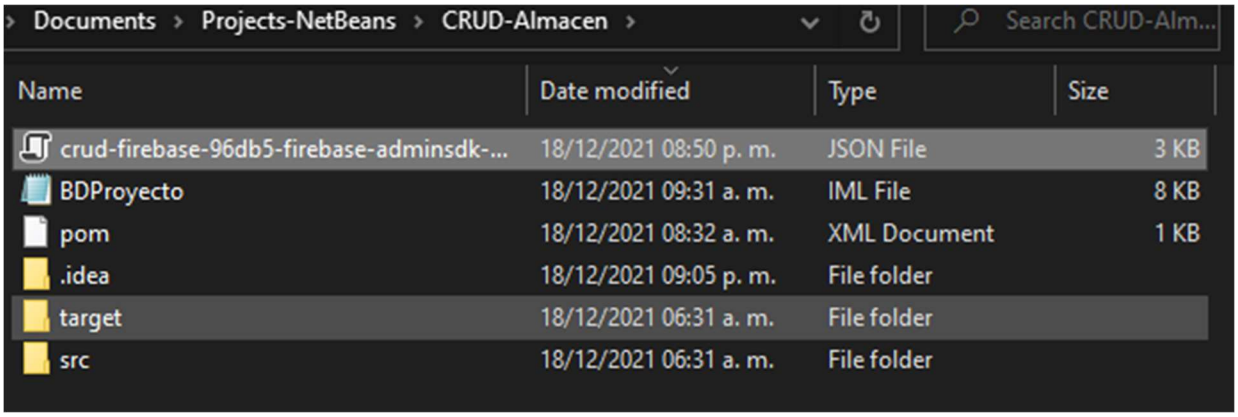




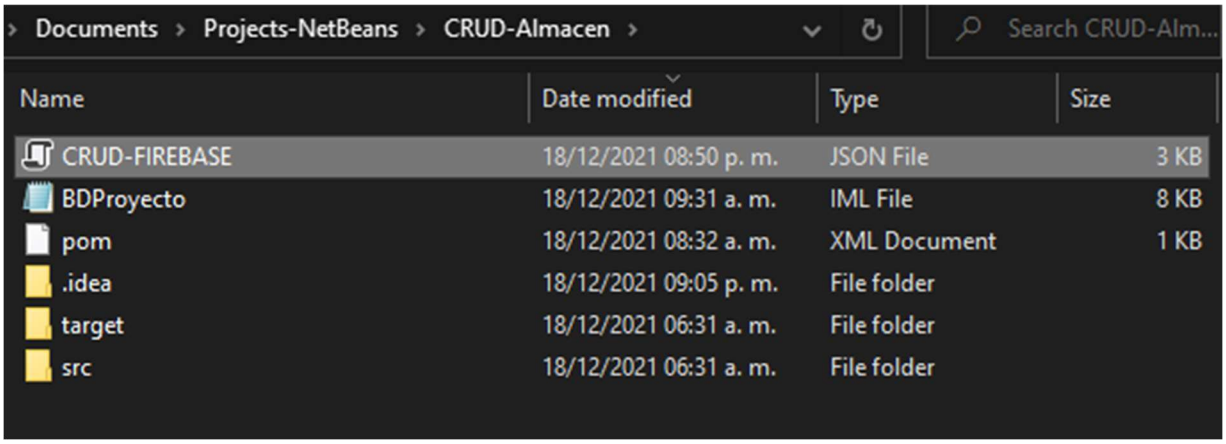


Una vez descargada la pegamos en el directorio raíz de nuestro proyecto de Java:

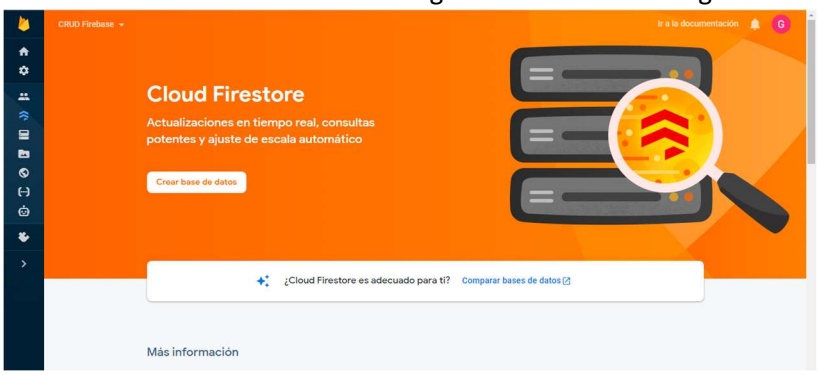




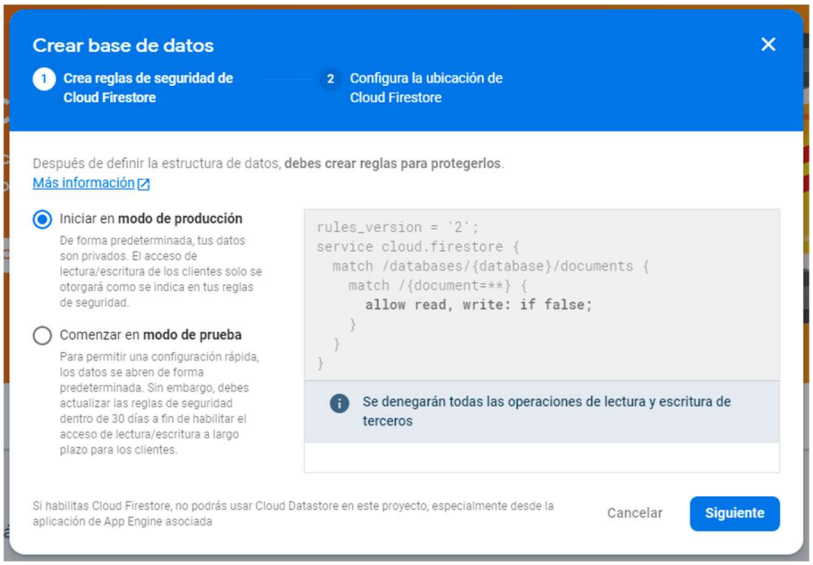
La renombramos a un nombre más corto:

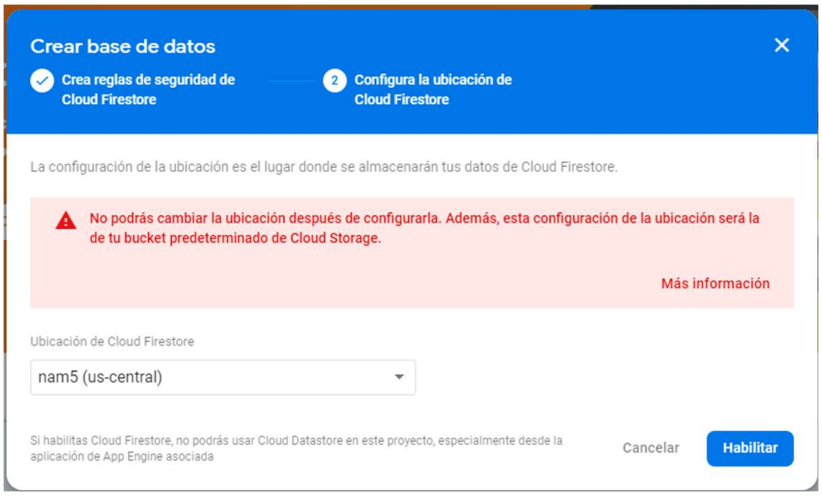


Creamos la base de datos donde se guardaran todos los registros:

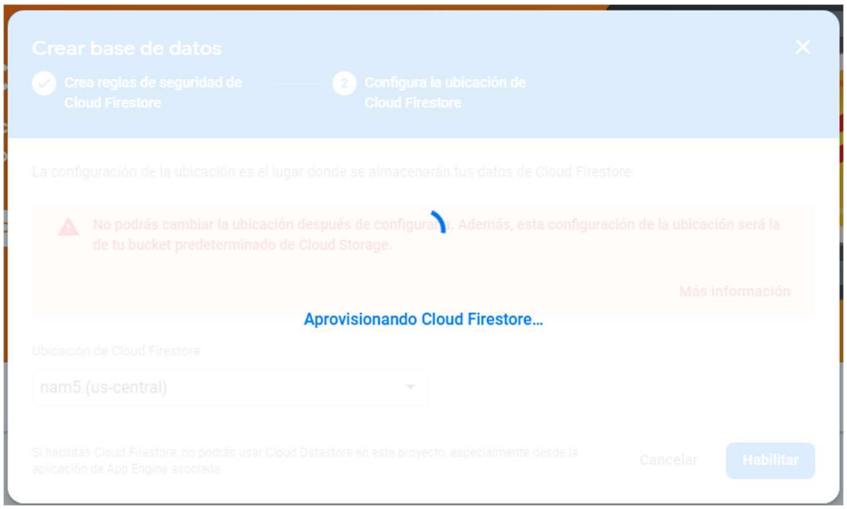


La creamos en modo de producción:

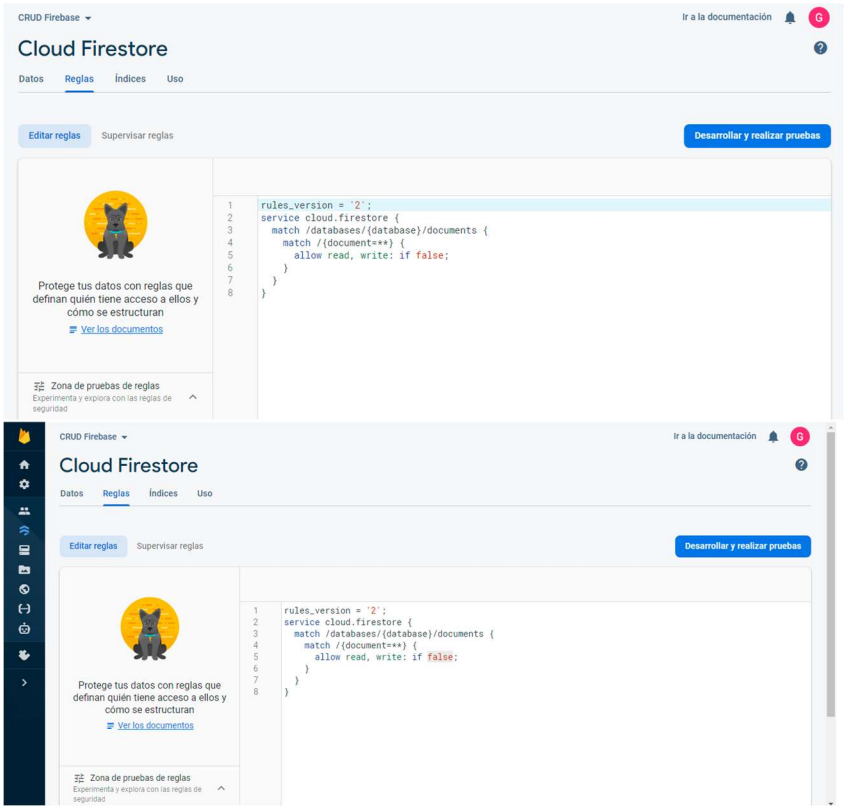




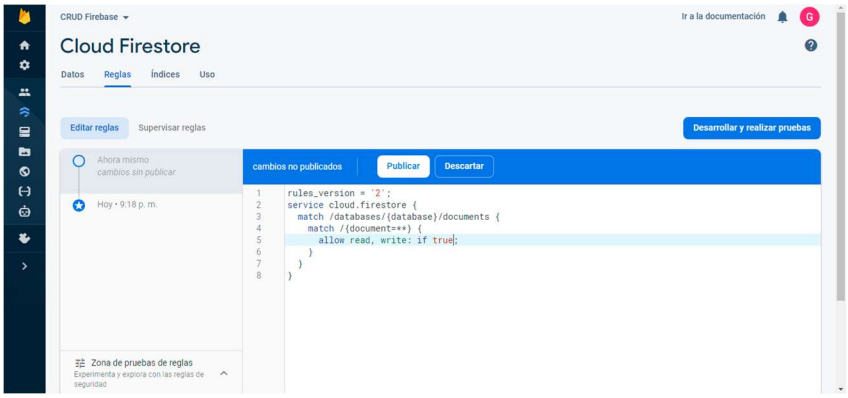
Una vez especificado el modo comienza a crear la BD:



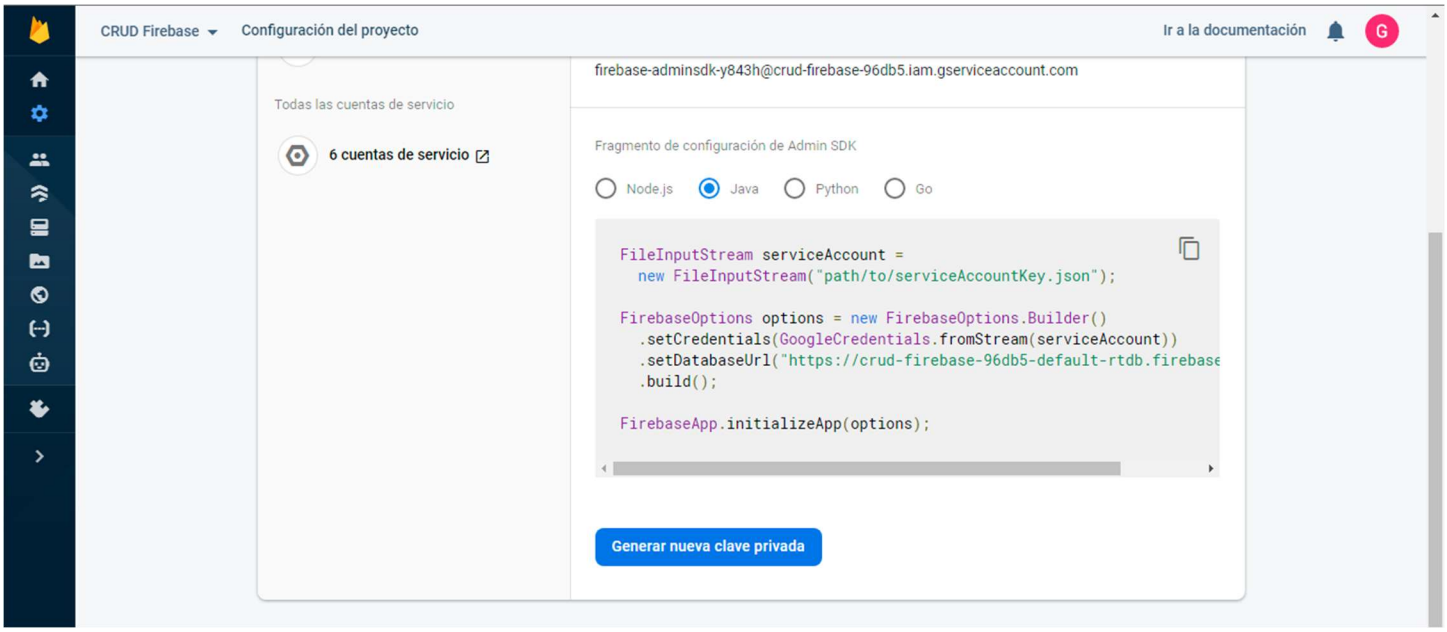
Ahora nos dirigimos a la pestaña de “Reglas” donde modificaremos el permiso de escritura:



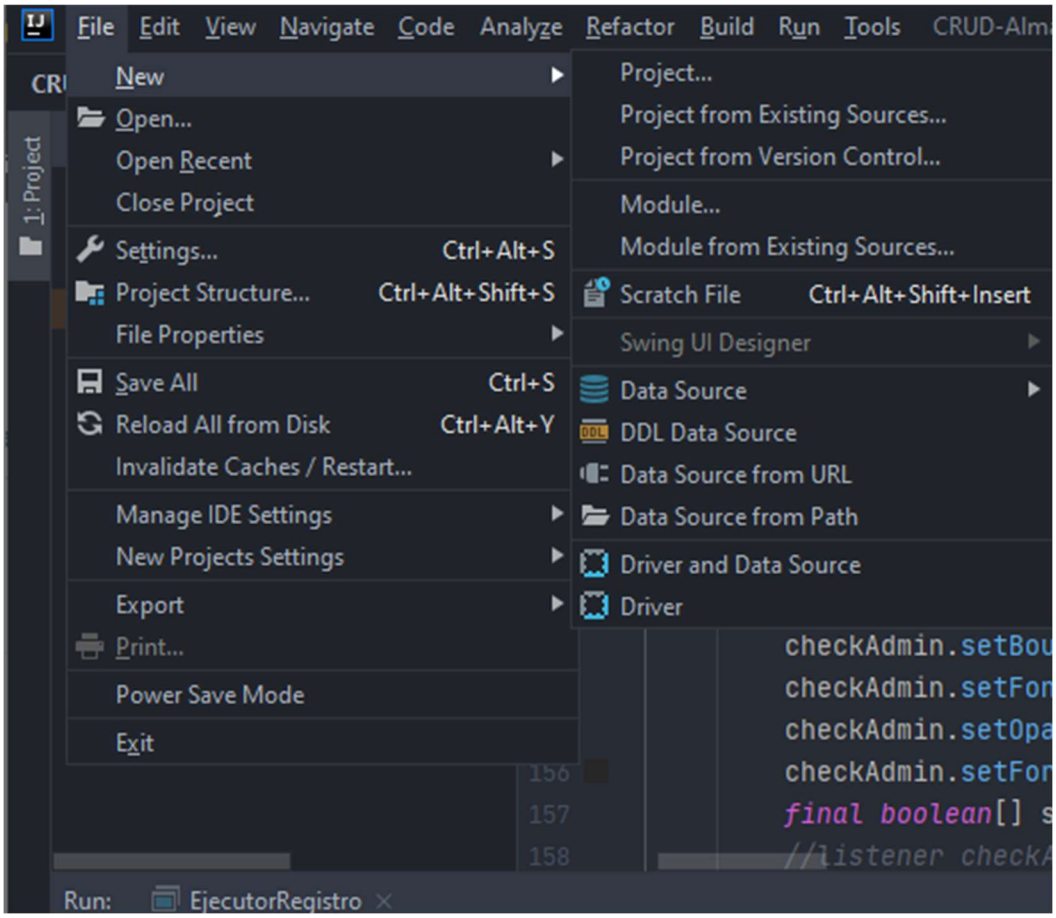
Modificamos la regla a “True “ para que nos permita los cambios:

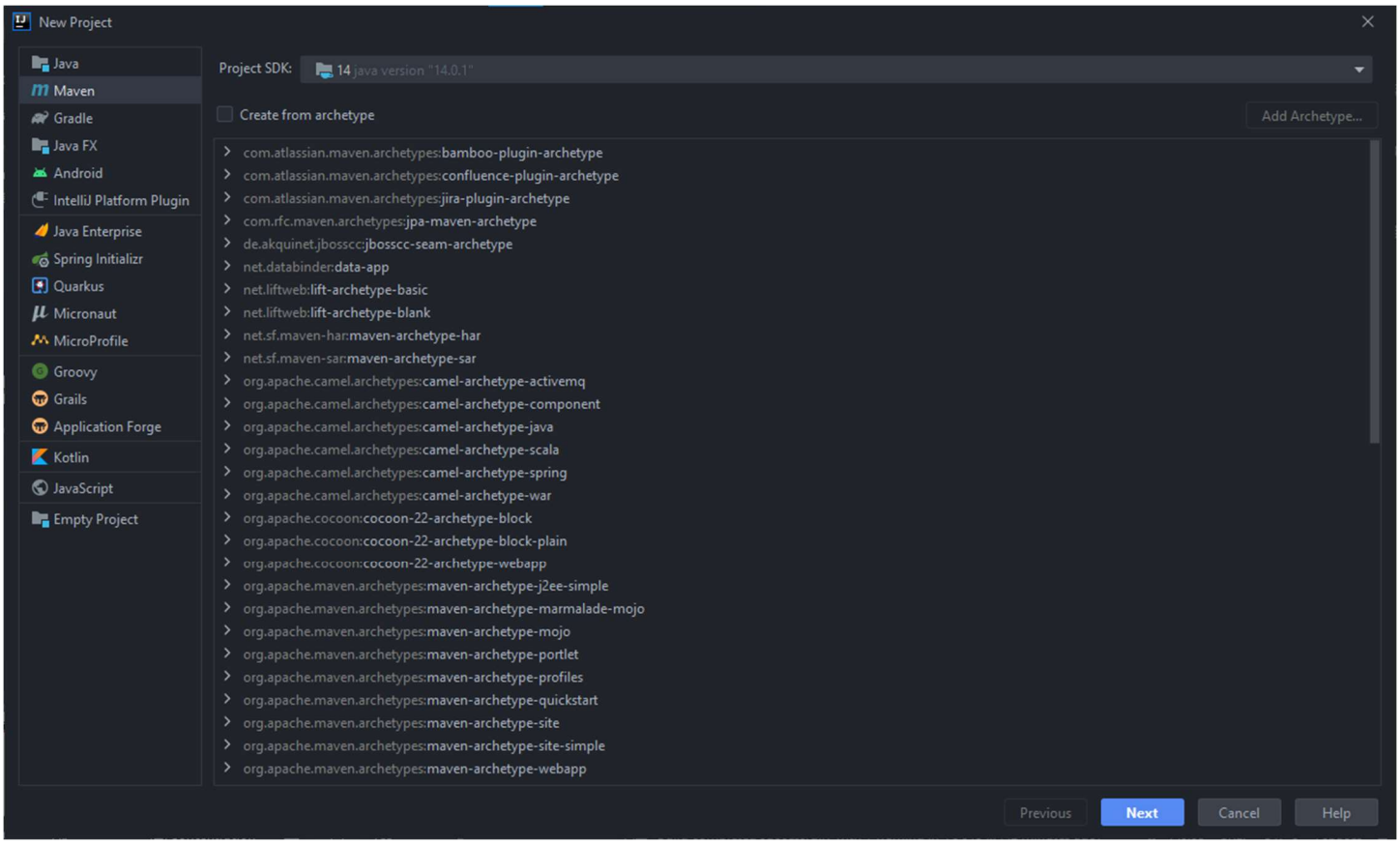


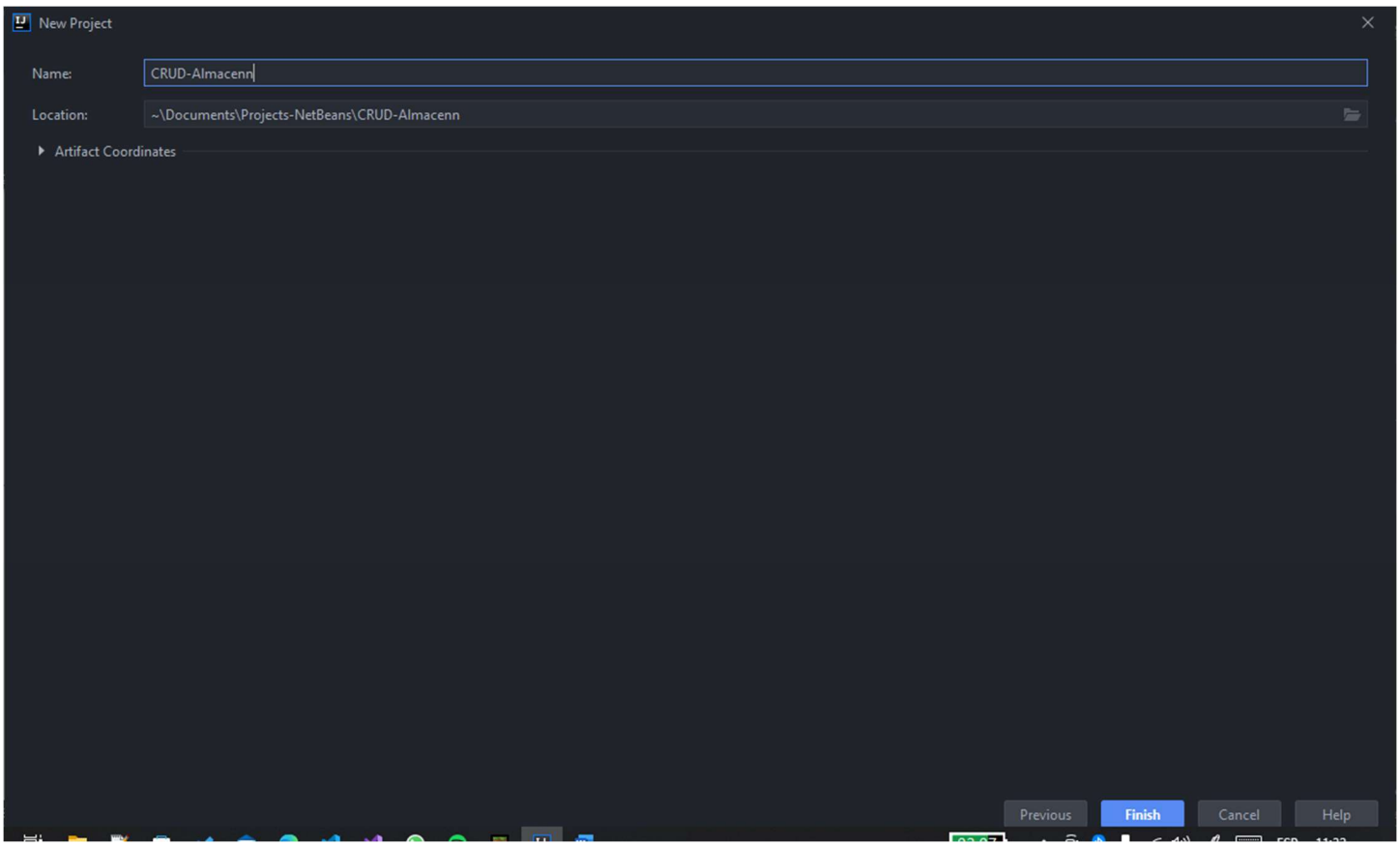
Después nos dirigimos a la sección de configuración del proyecto y en la pestaña de “Cuentas de servicio” elegiremos el SDK a usar en el proyecto, en este caso seleccionamos java y copiamos el código para realizar la conexión.



Ahora creamos un nuevo proyecto en intellij IDEA pero esta será un proyecto con Maven:







Finalizamos y empezamos la importación de dependencias mediante el archivo “pom.xml”

pom.xml (BDProyecto)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLoca
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mycompany</groupId>
    <artifactId>BDProyecto</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>
    <dependencies>
        <dependency>
            <groupId>com.google.firebase</groupId>
            <artifactId>firebase-admin</artifactId>
            <version>8.1.0</version>
        </dependency>
    </dependencies>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>14</maven.compiler.source>
        <maven.compiler.target>14</maven.compiler.target>
    </properties>
</project>
```

Una vez instaladas empezamos con la creación de interfaces graficas usando Java

Codigo del proyecto:

BorderLineRound.java

```java
import java.awt.Color;
import java.awt.Component;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.Shape;
import java.awt.geom.Path2D;
import java.awt.geom.RoundRectangle2D;
import javax.swing.border.AbstractBorder;


public class BorderLineRound extends AbstractBorder{

    public BorderLineRound(Color lineColor,boolean roundedCorners){

        this.lineColor = lineColor;
        this.roundedCorners = roundedCorners;

    }

    public void paintBorder(Component c,Graphics g,int x,int y,int width,int height){

        Graphics2D g2d = (Graphics2D) g;
        Shape outer;
        Shape inner;

        //lineaaaas

        if(roundedCorners){
```

```
            int offs = 1;
            int size = offs + offs;

            float arc = .2f * offs;

            g2d.setColor(lineColor);
            outer = new RoundRectangle2D.Float(x + 1, y + 1, width - 2 , height -2 ,offs
* 30, offs * height);
            inner = new RoundRectangle2D.Float(x + offs - 2, y + offs -2, width - size +
4, height - size + 4, arc, arc);
            Path2D path = new Path2D.Float(Path2D.WIND_EVEN_ODD);
            g2d.addRenderingHints(antiliasing);
            path.append(outer,false);
            path.append(inner,false);
            g2d.fill(path);

        }

        Color oldColor = c.getParent().getBackground();
        g2d.setColor(oldColor);

        int offs = 1;
        int size = offs + offs;
        float arc = .2f * offs;
        outer = new  RoundRectangle2D.Float(x,y,width,height,offs * 30, offs * height);
        inner = new RoundRectangle2D.Float(x + offs -2, y + offs - 2, width - size + 4,
height - size + 4 , arc, arc );
        Path2D path = new Path2D.Float(Path2D.WIND_EVEN_ODD);
        g2d.addRenderingHints(antiliasing);
        path.append(outer, false);
        path.append(inner, true);
        g2d.fill(path);
    }
    Color lineColor = null;
    boolean roundedCorners = false;

    RenderingHints antiliasing = new
RenderingHints(RenderingHints.KEY_ANTIALIASING,RenderingHints.VALUE_ANTIALIAS_ON);
}
```

JTextFieldRounded.java

```
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.event.FocusEvent;
import java.awt.event.FocusListener;
import javax.swing.JTextField;


public class JTextFieldRounded extends JTextField {
    private final Dimension d = new Dimension(250, 28);
    private final BorderLineRound border = new BorderLineRound((Color.lightGray), true);


    public JTextFieldRounded() {
        setOpaque(true);
        setBorder(border);
        setSize(d);
        setPreferredSize(d);
        setHorizontalAlignment(CENTER);
        setFont(new Font("Century Gothic", 0, 12));
```

```
        addFocusListener(new FocusListener() {

            public void focusGained(FocusEvent e) {
                txtFocusGained(e);
            }

            public void focusLost(FocusEvent e) {
                txtFocusLost(e);
            }

        });
    }


    private void txtFocusGained(FocusEvent evt) {

        setBorder(new BorderLineRound(Color.lightGray, true));
    }

    private void txtFocusLost(FocusEvent evt) {

        setBorder(border);
    }


}
```

RoundedBorder.java

```
import java.awt.Component;
import java.awt.Graphics;
import java.awt.Insets;
import javax.swing.border.Border;

class RoundedBorder implements Border {

    private int radius;

    RoundedBorder(int radius) {
        this.radius = radius;
    }

    public Insets getBorderInsets(Component c) {
        return new Insets(this.radius+1, this.radius+1, this.radius+2, this.radius);
    }

    public boolean isBorderOpaque() {
        return true;
    }

    public void paintBorder(Component c, Graphics g, int x, int y, int width, int height)
{
        g.drawRoundRect(x, y, width-1, height-1, radius, radius);
    }
}
```

Principal.java

```
import com.google.api.core.ApiFuture;
import com.google.auth.oauth2.GoogleCredentials;
import com.google.cloud.firestore.CollectionReference;
```

```
import com.google.cloud.firestore.DocumentSnapshot;
import com.google.cloud.firestore.Firestore;
import com.google.cloud.firestore.QuerySnapshot;
import com.google.firebase.FirebaseApp;
import com.google.firebase.FirebaseOptions;
import com.google.firebase.cloud.FirestoreClient;
import javax.imageio.ImageIO;
import javax.swing.*;
import java.awt.*;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.ExecutionException;

public class Principal extends JFrame {
    static Firestore bd;
    static String[] datosAcceso;

    public Principal() throws HeadlessException {
        this.setTitle("Inicio de Sesion");
        this.setSize(510, 470);
        this.setLocationRelativeTo(null);
        try {
            conectar();
            BDatos BDregistros = new BDatos();
            try {
                BDregistros.buscarUsuario();
            } catch (InterruptedException | ExecutionException interruptedException) {
                interruptedException.printStackTrace();
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Error en la conexión");
        }
        inicializarComponentes();
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    private void inicializarComponentes() {
        //Panel de Login
        JPanel panel = new JPanel();
        getContentPane().add(panel);
        panel.setLayout(null);

        //Letra para formulario
        Font letraGeneral = new Font("Century Gothic", Font.PLAIN, 28);
        //Etiqueta Nombre Formulario
        JLabel etiquetaSuperior = new JLabel("Bienvenido");
        etiquetaSuperior.setFont(letraGeneral);
        etiquetaSuperior.setHorizontalAlignment(JLabel.CENTER);
        etiquetaSuperior.setForeground(new Color(38, 38, 38));
        etiquetaSuperior.setBounds(150, 20, 200, 32);
        panel.add(etiquetaSuperior);
        //Imagen superior
        BufferedImage img3 = null;
        try {
            img3 = ImageIO.read(new
File("C:\\Users\\jesus\\Downloads\\RecursosBD\\carrito.png"));
        } catch (IOException e) {
            e.printStackTrace();
        }
```

```
assert img3 != null;
Image dimensionarIMG3 = img3.getScaledInstance(170, 160, Image.SCALE_SMOOTH);
ImageIcon imageIcon3 = new ImageIcon(dimensionarIMG3);

JLabel imagenSup = new JLabel(imageIcon3);
imagenSup.setBounds(158, 70, 170, 160);
panel.add(imagenSup);


JLabel Usuario, Contrasena;
Font letra = new Font("Century Gothic", Font.PLAIN, 23);
Font letra2 = new Font("Century Gothic", Font.PLAIN, 15);
//Etiqueta Usuario
Usuario = new JLabel("Usuario:");
Usuario.setBounds(100, 250, 150, 30);
Usuario.setForeground(new Color(38, 38, 38));
Usuario.setOpaque(false);
Usuario.setFont(letra);
panel.add(Usuario);
//panel.add(round);
JTextField round3 = new JTextField("");
round3.setBounds(100, 281, 300, 38);
round3.setFont(letra2);
round3.setOpaque(false);
round3.setForeground(new Color(38, 38, 38));
round3.setBackground(new Color(0, 0, 0));
round3.setBorder(new RoundedBorder(10));
panel.add(round3);
//Etiqueta Contrasena
Contrasena = new JLabel("Contraseña:");
Contrasena.setBounds(100, 320, 150, 30);
Contrasena.setForeground(new Color(38, 38, 38));
Contrasena.setOpaque(false);
Contrasena.setFont(letra);
panel.add(Contrasena);
//panel.add(round);
Font know = new Font("Webdings", Font.PLAIN, 15);
JTextField round4 = new JTextField("");
round4.setBounds(100, 350, 300, 38);
round4.setFont(know);
round4.setOpaque(false);
round4.setForeground(new Color(38, 38, 38));
round4.setBackground(new Color(0, 0, 0));
round4.setBorder(new RoundedBorder(10));
panel.add(round4);
System.out.println(round4.getText());

//Boton registro
JButton entrar = new JButton("Entrar");
entrar.setBounds(180, 400, 150, 25);
entrar.setForeground(new Color(38, 38, 38));
entrar.setFont(letra2);
entrar.setOpaque(false);
entrar.setFocusable(false);
entrar.setBorder(new RoundedBorder(20));
entrar.setBackground(new Color(0, 5, 8));
entrar.addActionListener(e -> {
    String usuario = round3.getText();
    String pass = round4.getText();
    String total = usuario + "-" + pass;
    boolean existe = false;
    for (String s : datosAcceso) {
```

```
                if (s.equals(total)) {
                    existe = true;
                    break;
                } else
                    existe = false;
            }

            if (existe) {
                JOptionPane.showMessageDialog(null, "Inicio de sesion exitoso");
                System.exit(0);
            } else {
                JOptionPane.showMessageDialog(null, "Error en credenciales o
registrese");
                round3.setText(null);
                round4.setText(null);
            }

        });

        //Panel Secundario
        JPanel panel2 = new JPanel();
        panel.setLayout(null);


        panel.getRootPane().setDefaultButton(entrar);
        panel.add(entrar);

    }

    private void conectar() throws IOException {

        FileInputStream serviceAccount =
                new FileInputStream("CRUD-FIREBASE.json");

        FirebaseOptions options = new FirebaseOptions.Builder()
                .setCredentials(GoogleCredentials.fromStream(serviceAccount))
                .setDatabaseUrl("https://crud-firebase-96db5-default-
rtdb.firebaseio.com")
                .build();

        FirebaseApp.initializeApp(options);

        bd = FirestoreClient.getFirestore();

        System.out.println("Se realizo la conexion con exito");

    }

    static class Datos {
        private final String Name;
        private final String Usuario;
        private final String Password;
        private final String Admin;

        public Datos(String name, String usuario, String password, String admin) {
            Name = name;
            Usuario = usuario;
            Password = password;
            Admin = admin;
        }

        public String getName() {
            return Name;
```

```
        }

        public String getAdmin() {
            return Admin;
        }

        public String getUsuario() {
            return Usuario;
        }

        public String getPassword() {
            return Password;
        }
    }

    static class BDatos {
        List<Datos> listaRegistros = new ArrayList<>();

        public void buscarUsuario() throws InterruptedException, ExecutionException {
            CollectionReference Registros = bd.collection("Registro");
            ApiFuture<QuerySnapshot> querySnapshot = Registros.get();

            for (DocumentSnapshot document : querySnapshot.get().getDocuments()) {
                Datos registros = new Datos(document.getString("Nombre"),
document.getString("Usuario"),
                        document.getString("Password"), document.getString("Admin"));

                listaRegistros.add(registros);
            }
            mostrarRegistros();
        }

        public void mostrarRegistros() {
            Datos[] dat = new Datos[listaRegistros.size()];
            String[] datPrueba = new String[listaRegistros.size()];
            for (int i = 0; i < listaRegistros.size(); i++) {
                dat[i] = listaRegistros.get(i);
                //datPrueba[i] = dat[i].getUsuario() + "-" + dat[i].getPassword();
                datPrueba[i] = dat[i].getName() + " - " +dat[i].getUsuario() + " - " +
dat[i].getPassword()+ " - " +dat[i].getAdmin();
            }

            for (String s : datPrueba) {
                System.out.println(s);
            }
            datosAcceso = datPrueba;
        }
    }

}
```

Registro.java

```
import com.google.api.core.ApiFuture;

import javax.imageio.ImageIO;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.image.BufferedImage;
import java.io.File;
```

```
import java.io.IOException;

import com.google.auth.oauth2.GoogleCredentials;
import com.google.cloud.firestore.DocumentReference;
import com.google.cloud.firestore.Firestore;
import com.google.cloud.firestore.WriteResult;
import com.google.firebase.cloud.FirestoreClient;
import com.google.firebase.FirebaseApp;
import com.google.firebase.FirebaseOptions;

import java.io.FileInputStream;
import java.util.HashMap;
import java.util.Map;

public class Registro extends JFrame {
    static Firestore bd;

    public Registro() throws HeadlessException {
        this.setTitle("Registro");
        this.setSize(500, 620);
        this.setLocationRelativeTo(null);
        //this.setUndecorated(true);
        //this.setBackground(new Color(0, 0, 0, 230));
        try {
            conectar();
        } catch (Exception e) {
            e.getStackTrace();
        }
        inicializarComponentes();
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        //this.setShape(new RoundRectangle2D.Double(0, 0, getWidth(), getHeight(), 20,
20));
    }

    private void inicializarComponentes() {
        JPanel panel = new JPanel();
        getContentPane().add(panel);
        //panel.setOpaque(false);
        panel.setLayout(null);

        //Letra para formulario
        Font letraGeneral = new Font("Century Gothic", Font.PLAIN, 28);
        //Etiqueta Nombre Formulario
        JLabel etiquetaSuperior = new JLabel("Registro");
        etiquetaSuperior.setFont(letraGeneral);
        etiquetaSuperior.setHorizontalAlignment(JLabel.CENTER);
        etiquetaSuperior.setForeground(new Color(38, 38, 38));
        etiquetaSuperior.setBounds(170, 20, 145, 32);
        panel.add(etiquetaSuperior);
        //Imagen superior
        BufferedImage img3 = null;
        try {
            img3 = ImageIO.read(new
File("C:\\Users\\jesus\\Downloads\\RecursosBD\\carrito.png"));
        } catch (IOException e) {
            e.printStackTrace();
        }

        assert img3 != null;
        Image dimensionarIMG3 = img3.getScaledInstance(170, 160, Image.SCALE_SMOOTH);
        ImageIcon imageIcon3 = new ImageIcon(dimensionarIMG3);

        JLabel imagenSup = new JLabel(imageIcon3);
```

```
imagenSup.setBounds(158, 60, 170, 160);
panel.add(imagenSup);

//Etiqueta Nombre
JLabel Nombre, Correo, Usuario, Contrasena;
Nombre = new JLabel("Nombre:");
Font letra = new Font("Century Gothic", Font.PLAIN, 23);
Font letra2 = new Font("Century Gothic", Font.PLAIN, 15);

Nombre.setBounds(100, 250, 150, 30);
Nombre.setForeground(new Color(38, 38, 38));
Nombre.setOpaque(false);
Nombre.setFont(letra);
panel.add(Nombre);
//Cuadro de texto

//panel.add(round1);
JTextField round = new JTextField("");
round.setBounds(100, 281, 300, 38);
round.setFont(letra2);
round.setOpaque(false);
round.setHorizontalAlignment(JTextField.CENTER);
round.setForeground(new Color(38, 38, 38));
round.setBackground(new Color(0, 0, 0, 10));
round.setBorder(new RoundedBorder(10));
panel.add(round);

/*//Etiqueta Correo
Correo = new JLabel("Correo:");
Correo.setBounds(100, 320, 150, 30);
Correo.setForeground(new Color(38, 38, 38));
Correo.setOpaque(false);
Correo.setFont(letra);
panel.add(Correo);
//panel.add(round);
JTextField round2 = new JTextField("");
round2.setBounds(100, 350, 300, 38);
round2.setFont(letra2);
round2.setOpaque(false);
round2.setForeground(new Color(38, 38, 38));
round2.setBackground(new Color(0, 0, 0, 10));
round2.setBorder(new RoundedBorder(10));
panel.add(round2);
*/

//Etiqueta Usuario
Usuario = new JLabel("Usuario:");
Usuario.setBounds(100, 320, 150, 30);
Usuario.setForeground(new Color(38, 38, 38));
Usuario.setOpaque(false);
Usuario.setFont(letra);
panel.add(Usuario);
//panel.add(round);
JTextField round3 = new JTextField("");
round3.setBounds(100, 350, 300, 38);
round3.setFont(letra2);
round3.setHorizontalAlignment(JTextField.CENTER);
round3.setOpaque(false);
round3.setForeground(new Color(38, 38, 38));
round3.setBackground(new Color(0, 0, 0, 10));
round3.setBorder(new RoundedBorder(10));
panel.add(round3);
//Etiqueta Contrasena
```

```
        Contrasena = new JLabel("Contraseña:");
        Contrasena.setBounds(100, 390, 150, 30);
        Contrasena.setForeground(new Color(38, 38, 38));
        Contrasena.setOpaque(false);
        Contrasena.setFont(letra);
        panel.add(Contrasena);
        //panel.add(round);
        Font know = new Font("Webdings", Font.PLAIN, 15);
        JTextField round4 = new JTextField("");
        round4.setBounds(100, 420, 300, 38);
        round4.setFont(letra2);
        round4.setOpaque(false);
        round4.setForeground(new Color(38, 38, 38));
        round4.setBackground(new Color(0, 0, 0, 10));
        round4.setBorder(new RoundedBorder(10));
        panel.add(round4);
        System.out.println(round4.getText());
        //Division Admin
        JCheckBox checkAdmin = new JCheckBox("Administrador");
        checkAdmin.setBounds(100, 470, 300, 30);
        checkAdmin.setFont(letra2);
        checkAdmin.setOpaque(false);
        checkAdmin.setForeground(new Color(38, 38, 38));
        final boolean[] statAdmin = {false};
        //listener checkAdmin
        checkAdmin.addItemListener(new ItemListener() {
            @Override
            public void itemStateChanged(ItemEvent e) {
                System.out.println(e.getStateChange());
                if (e.getStateChange() == 1) {
                    System.out.println("check");
                    statAdmin[0] = true;
                } else {
                    System.out.println("Uncheck");
                    statAdmin[0] = false;
                }
                System.out.println(e.getStateChange() + " final");
                System.out.println(statAdmin[0]);
            }
        });
        panel.add(checkAdmin);

        //Imagen Hide
        BufferedImage hideimg = null;
        try {
            hideimg = ImageIO.read(new File("C:\\Users\\jesus\\Documents\\Projects-
NetBeans\\UniversidadParcial2\\src\\Imagenes\\hide.png"));
        } catch (IOException e) {
            e.printStackTrace();
        }

        assert hideimg != null;
        Image dimensionarHide = hideimg.getScaledInstance(35, 35, Image.SCALE_SMOOTH);
        ImageIcon imageIconHide = new ImageIcon(dimensionarHide);
        //Imagen Show
        BufferedImage showimg = null;
        try {
            showimg = ImageIO.read(new File("C:\\Users\\jesus\\Documents\\Projects-
NetBeans\\UniversidadParcial2\\src\\Imagenes\\show.png"));
        } catch (IOException e) {
            e.printStackTrace();
        }
```

```
        assert showimg != null;
        Image dimensionarShow = showimg.getScaledInstance(38, 38, Image.SCALE_SMOOTH);
        ImageIcon imageIconShow = new ImageIcon(dimensionarShow);

        //Boton Password
        JButton hideShow = new JButton();
        hideShow.setIcon(imageIconHide);
        hideShow.setBounds(400, 490, 38, 38);
        hideShow.setBorder(null);
        hideShow.setOpaque(false);
        hideShow.setBackground(new Color(0, 0, 1, 100));
        panel.add(hideShow);
        //Boton registro
        JButton registro = new JButton("Registrar");
        registro.setBounds(180, 540, 150, 25);
        registro.setForeground(new Color(38, 38, 38));
        registro.setFont(letra2);
        registro.setOpaque(false);
        registro.setFocusable(false);
        registro.setBorder(new RoundedBorder(20));
        registro.setBackground(new Color(0, 5, 8, 12));

        registro.addActionListener(e -> {
            String Nombre1 = round.getText();
            String Usuario1 = round3.getText();
            String Pass = round4.getText();
            String adminDude = String.valueOf(statAdmin[0]);
            Map<String, Object> datosRegistro = new HashMap<>();
            datosRegistro.put("Nombre", Nombre1);
            //datosRegistro.put("Correo", Correo1);
            datosRegistro.put("Usuario", Usuario1);
            datosRegistro.put("Password", Pass);
            datosRegistro.put("Admin",adminDude);
            String uuid = java.util.UUID.randomUUID().toString();
            insertarDatos("Registro", uuid, datosRegistro);

            round.setText(null);
            //round2.setText(null);
            round3.setText(null);
            round4.setText(null);


        });

        panel.add(registro);
        panel.getRootPane().setDefaultButton(registro);

    }

    private void conectar() throws IOException {


        FileInputStream serviceAccount =
                new FileInputStream("CRUD-FIREBASE.json");

        FirebaseOptions options = new FirebaseOptions.Builder()
                .setCredentials(GoogleCredentials.fromStream(serviceAccount))
                .setDatabaseUrl("https://crud-firebase-96db5-default-
rtdb.firebaseio.com")
                .build();

        FirebaseApp.initializeApp(options);
```

```
        /*FileInputStream serviceAccount
                = new FileInputStream("ordinarioBD.json");

        FirebaseOptions options = new FirebaseOptions.Builder()
                .setCredentials(GoogleCredentials.fromStream(serviceAccount))
                .setDatabaseUrl("https://ordinariobd-default-rtdb.firebaseio.com")
                .build();

        FirebaseApp.initializeApp(options);

         */

        bd = FirestoreClient.getFirestore();

        System.out.println("Se realizo la conexion con exito");

    }

    public static boolean insertarDatos(
            String coleccion,
            String documento,
            Map<String, Object> data) {
        try {
            DocumentReference docRef = bd.collection(coleccion).document(documento);
            ApiFuture<WriteResult> result = docRef.set(data);
            System.out.println("Hora de actualizacion: " + result.get().getUpdateTime());

            return true;
        } catch (Exception e) {
            e.getStackTrace();
        }
        return false;
    }
}
```

EjecutorPrincipal.java

```
public class EjecutorPrincipal {
    public static void main(String[] args) {
        Principal logon = new Principal();
        logon.setVisible(true);
    }
}
```

EjecutorRegistro.java

```
public class EjecutorRegistro {
    public static void main(String[] args) {
        Registro window = new Registro();
        window.setVisible(true);
    }
}
```

Salidas de pantalla:

Enlace a sitio GitHub del proyecto:

GaryElizalde224/Extraordinario-BaseDeDatos (github.com)