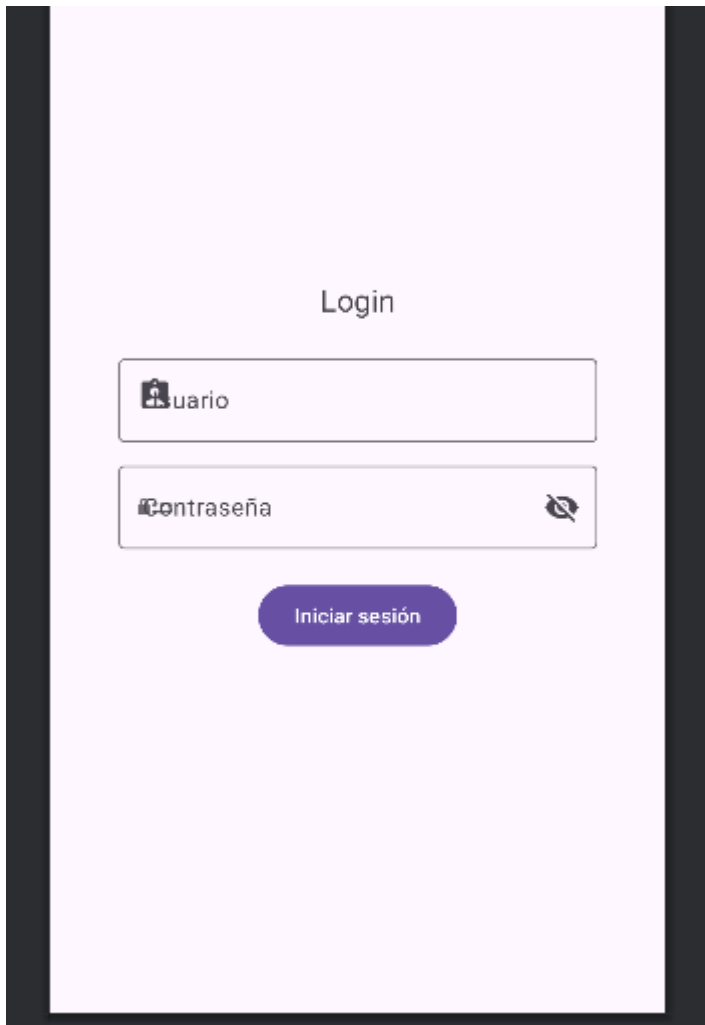


Creamos layout Login

(No olvidar crear la conexión a firebase)



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    tools:context=".Login">

    <TextView
        android:id="@+id/tvLogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"
```

```

        android:textSize="20sp"
        android:layout_marginBottom="20dp"/>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/tiUsuario"
    android:layout_width="320dp"
    android:layout_height="wrap_content"
    app:startIconDrawable="@drawable/ic_user"
    app:endIconMode="clear_text"
    android:layout_marginBottom="10dp">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/etUsuario"
        android:layout_width="320dp"
        android:layout_height="wrap_content"
        android:hint="Usuario"/>

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/tiPassword"
    android:layout_width="320dp"
    android:layout_height="wrap_content"
    app:startIconDrawable="@drawable/ic_password"
    app:endIconMode="password_toggle"
    android:layout_marginBottom="20dp">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/etPassword"
        android:layout_width="320dp"
        android:layout_height="wrap_content"
        android:hint="Contraseña"
        android:inputType="text"/>

</com.google.android.material.textfield.TextInputLayout>

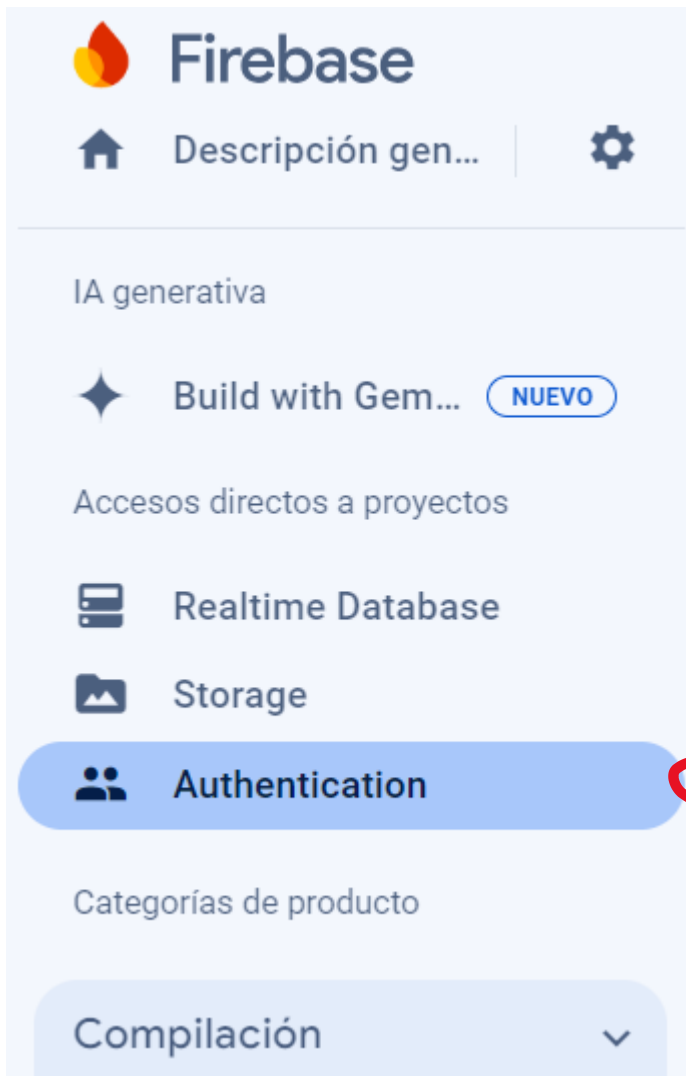
<Button
    android:id="@+id/btnLogin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Iniciar sesión"
    android:onClick="onLoginClick"/>

</LinearLayout>

```

Creamos layout Login

Activar Authentication



Authentication

Usuarios Método de acceso Plantillas Uso Configuración Extensions

Proveedores de acceso

Agrega tu primer método de acceso y comienza a utilizar Firebase Auth

Proveedores nativos

Correo electrónico/contraseña

Teléfono

Anónimo

Proveedores adicionales

Google

Facebook

Play Juegos

Game Center

Apple

GitHub

Microsoft

Twitter

Vahoo

Proveedores personalizados

OpenID Connect

SAML

Proveedores de acceso

Correo electrónico/contraseña

Habilitar

Permite que los usuarios se registren con su dirección de correo electrónico y contraseña. Nuestros SDK también proporcionan verificación de la dirección de correo electrónico, recuperación de contraseñas y primitivas de cambio de dirección de correo. [Más información](#)

Vínculo del correo electrónico (acceso sin contraseña)

Habilitar

Cancelar

Guardar

Authentication

[Usuarios](#) [Método de acceso](#) [Plantillas](#) [Uso](#) [Configuración](#) [Extensions](#)

⚠ El acceso con redireccionamiento de origen cruzado en Google Chrome M115 y versiones posteriores ya no es compatible y dejará de funcionar el 24 de junio de 2024. ✓

Agregar usuario



Identificador	Proveedores	Fecha de creación ↓	Fecha de acceso	UID de usuario
---------------	-------------	---------------------	-----------------	----------------

Este proyecto todavía no tiene usuarios

⚠ El acceso con redireccionamiento de origen cruzado en Google Chrome M115 y versiones posteriores ya no es compatible y dejará de funcionar el 24 de junio de 2024. ✓

Agregar usuario



Identificador	Proveedores	Fecha de creación ↓	Fecha de acceso	UID de usuario
---------------	-------------	---------------------	-----------------	----------------

Agrega un usuario con correo electrónico y contraseña

Correo electrónico

admin@admin.cl

Contraseña

1234

Cancelar

Agregar usuario

Agregamos las variables init una para binding y otra para el auth de firebase

```
class Login : AppCompatActivity() {  
    private lateinit var binding: ActivityLoginBinding  
    private lateinit var firebaseAuth: FirebaseAuth  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        binding = ActivityLoginBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
  
        firebaseAuth = FirebaseAuth.getInstance()  
    }  
}
```

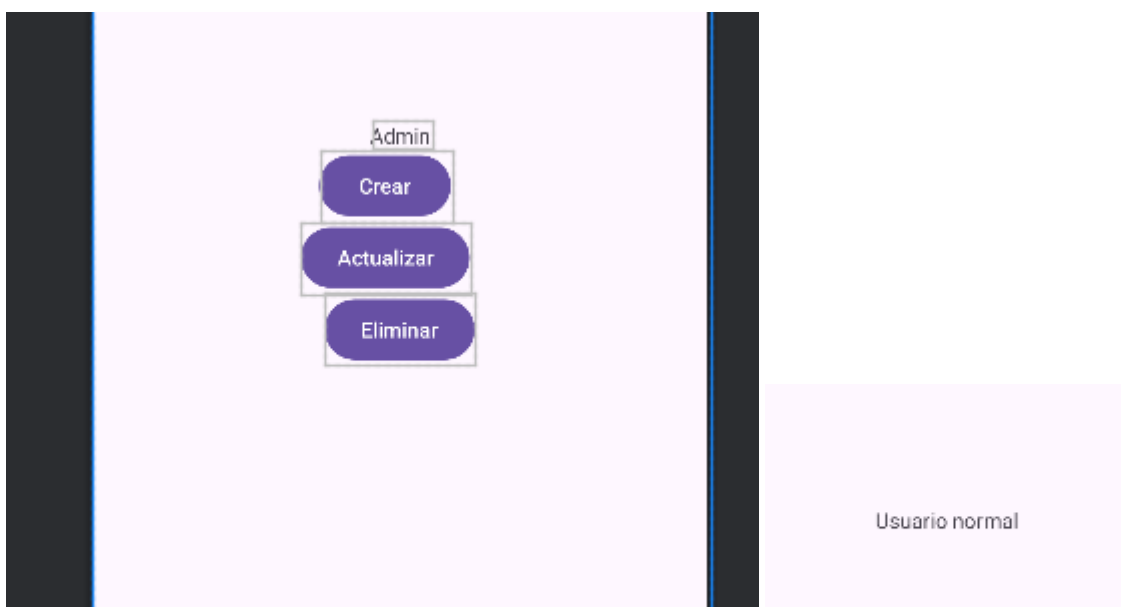
Comprobamos si encuentra match en la base de datos si lo hace incia los intents

```
binding.btnLogin.setOnClickListener { it: View!
    val inicio = binding.etUsuario.text.toString()
    val pass = binding.etPassword.text.toString()
    val almacenar = inicio.substringAfter( delimiter: "@" )

    firebaseAuth.signInWithEmailAndPassword(
        inicio, pass
    )

    .addOnCompleteListener { task ->
        if (almacenar == "admin.cl") {
            var intent = Intent( packageContext: this, AdminActivity::class.java )
            startActivity(intent)
        } else if (almacenar == "normal.cl") {
            var intent = Intent( packageContext: this, UsuarioNormalActivity::class.java )
            startActivity(intent)
        }
    }
}
```

Nos damos cuenta que debemos tener un activityAdmin y otro para usuarios normales



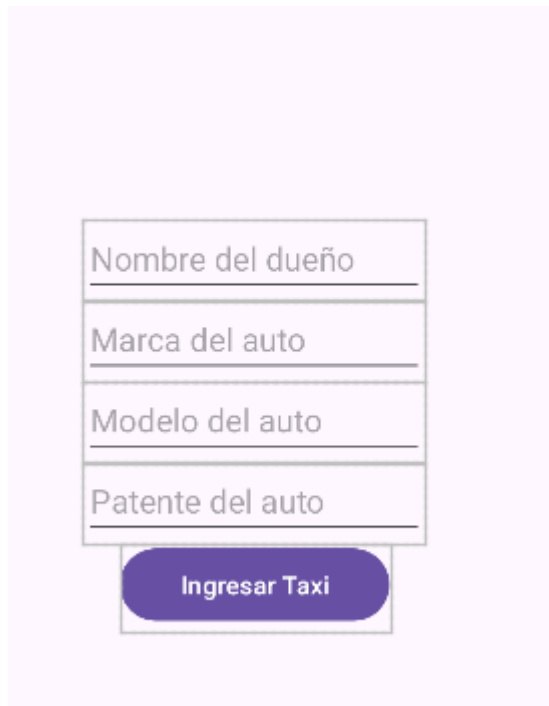
En las activitys de os menú se deben hacer los intents correspondiente que lleve a cada layout o actividad del crud

EMPEZAMOS CON REGISTRAR

Debemos crear nuestra dataclass que se encargará de almacenar los datos que iran a la base de datos

```
data class DataClass(val nombreDuenio:String?=null, val  
modeloAuto:String?=null, val marcaAuto:String?=null, val  
patenteAuto:String?=null)
```

Creamos el layout correspondiente GUI



```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:id="@+id/main"  
    android:layout_width="match_parent"
```

```

        android:layout_height="match_parent"
        tools:context=".CrearActivity"
        android:gravity="center"
        android:orientation="vertical">

        <EditText
            android:id="@+id/etNombreDuenio"
            android:layout_width="190dp"
            android:layout_height="wrap_content"
            android:hint="Nombre del dueño"></EditText>

        <EditText
            android:id="@+id/etMarcaAuto"
            android:layout_width="190dp"
            android:layout_height="wrap_content"
            android:hint="Marca del auto"></EditText>

        <EditText
            android:id="@+id/etModeloAuto"
            android:layout_width="190dp"
            android:layout_height="wrap_content"
            android:hint="Modelo del auto"></EditText>

        <EditText
            android:id="@+id/etPatenteAuto"
            android:layout_width="190dp"
            android:layout_height="wrap_content"
            android:hint="Patente del auto"></EditText>

        <Button
            android:id="@+id/btInsertarTaxi"
            android:layout_width="150dp"
            android:layout_height="wrap_content"
            android:text="Ingresar Taxi"
            ></Button>
    </LinearLayout>

```

En nuestra clase Crear Accedemos al botón de crear mediante binding y seteamos el ClickListener

```

class CrearActivity : AppCompatActivity() {
    private lateinit var binding: ActivityCrearBinding
    private lateinit var dataBaseReferences: DatabaseReference
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityCrearBinding.inflate(layoutInflater)
        setContentView(binding.root)

        dataBaseReferences=FirebaseDatabase.getInstance().getReference("Datos

```



```
Taxis")

        binding.btInsertarTaxi.setOnClickListener {
            val nombreDuenio = binding.etNombreDuenio.text.toString()
            val marcaAuto = binding.etMarcaAuto.text.toString()
            val modeloAuto = binding.etModeloAuto.text.toString()
            val patenteAuto = binding.etPatenteAuto.text.toString()
            val taxi =
DataClass(nombreDuenio, modeloAuto, marcaAuto, patenteAuto)
```

(no olvidar la referencia a firebase)

Dentro de nuestro listener rescatamos los valores de los inputs y creamos nuestro objeto Taxi (DataClass)

Dentro de ONCREATE evaluamos si nuestros campos estan vacíos o nulos

```
if (patenteAuto.isNullOrEmpty() || patenteAuto.isBlank()) {
    Toast.makeText(this, "Registro fallido", Toast.LENGTH_SHORT).show()
}
//Si está vacío muestra error ↑
else {
    val consultaPatente =
FirebaseDatabase.getInstance().getReference("Datos Taxis/$patenteAuto")
    consultaPatente.addListenerForSingleValueEvent(object :
ValueEventListener{
        override fun onDataChange(snapshot: DataSnapshot) {
            if (snapshot.exists()) {
                Toast.makeText(this@CrearActivity, "Error patente
$patenteAuto ya esta registrada", Toast.LENGTH_SHORT).show()
//Hasta aquí evalua si dato ya
existe
            }
            else{
                val dbRef =
dataBaseReferences.child(patenteAuto).setValue(taxi)
                dbRef.addListener {
                    Toast.makeText(this@CrearActivity, "Registro
exitoso", Toast.LENGTH_SHORT).show()
//Aquí ya el registro fue exitoso e
incia un intent hacia MainActivity
                val intent =
Intent(this@CrearActivity, MainActivity::class.java)
                startActivity(intent)
                finish()
            }
        }
    })
}
```

```

    }

    //Aquí escucha si el evento
    falló
    dbRef.addOnFailureListener {
        Toast.makeText(this@CrearActivity, "Registro
    fallido", Toast.LENGTH_SHORT).show()
    }
}

override fun onCancelled(error: DatabaseError) {
    Toast.makeText(this@CrearActivity, "Error en la búsqueda",
    Toast.LENGTH_SHORT).show()
}

})
}

```

LeerActivity y actualizar

Crear los layouts correspondientes

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:layout_margin="16dp"
    tools:context=".LeerActivity">

    <EditText
        android:id="@+id/EtBuscarPatenteAuto"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Ingrese Patente">

    </EditText>

    <Button
        android:id="@+id/BtBuscarDatos"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Buscar Patente"
    >

```

```
        android:layout_marginBottom="16dp">

</Button>
<EditText
    android:id="@+id/etNombreDuenio"
    android:layout_width="190dp"
    android:layout_height="wrap_content"
    android:hint="Nombre del dueño"></EditText>

<EditText
    android:id="@+id/etMarcaAuto"
    android:layout_width="190dp"
    android:layout_height="wrap_content"
    android:hint="Marca del auto"></EditText>

<EditText
    android:id="@+id/etModeloAuto"
    android:layout_width="190dp"
    android:layout_height="wrap_content"
    android:hint="Modelo del auto"></EditText>

<EditText
    android:id="@+id/etPatenteAuto"
    android:layout_width="190dp"
    android:layout_height="wrap_content"
    android:hint="Patente del auto"></EditText>

<Button
    android:id="@+id/BtGuardarDatos"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Guardar"
    android:layout_marginBottom="16dp">

</Button>
</LinearLayout>
```

Ingrese Patente

Buscar Patente

Nombre del dueño

Marca del auto

Modelo del auto

Patente del auto

Guardar

Lo de siempre crear las late init var setear los botones con listeners

```
class LeerActivity : AppCompatActivity() {  
    private lateinit var binding: ActivityLeerBinding  
    private lateinit var databaseReference: DatabaseReference  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        enableEdgeToEdge()  
        binding = ActivityLeerBinding.inflate(layoutInflater)  
        setContentView(R.layout.activity_leer)  
  
        binding.BtBuscarDatos.setOnClickListener {  
            val patente = binding.EtBuscarPatenteAuto.text.toString()  
  
            leerDatos(patente)  
        }  
    }  
}
```

el boton buscar datos se encargará de traerlos desde firebase con la función leerDatos(patente)

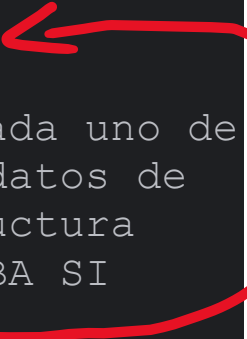
```
private fun leerDatos(patente: String) {
    databaseReference =
        FirebaseDatabase.getInstance().getReference("Datos Taxis")
    val data = databaseReference.child(patente).get()
    //Se traen los datos haciendo referencia al
    nodo ↑

    data.addListenerSuccess {
        if (it.exists()) {
            val nombreDuenio = it.child("nombreDuenio").value.toString()
            val modeloAuto = it.child("modeloAuto").value.toString()
            val marcaAuto = it.child("marcaAuto").value.toString()
            val patenteAuto = it.child("patenteAuto").value.toString()

            binding.etNombreDuenio.setText(nombreDuenio)
            binding.etModeloAuto.setText(modeloAuto)
            binding.etMarcaAuto.setText(marcaAuto)
            binding.etPatenteAuto.setText(patenteAuto)

            //se crean variable para almacenar cada uno de
            los atributos del objeto creado con datos de
            firebase (coinciden con nuestra estructura
            dataclass)AL MISMO TIEMPO SE COMPRUEBA SI
            EXISTEN↑. PD:SE SETEAN LOS CAMPOS
        } else {
            Toast.makeText(this, "Error: Patente no Registrada",
                Toast.LENGTH_SHORT).show()
            //Si no existen se muestra mensaje de error↑
        }
    }

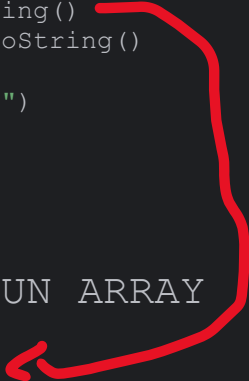
    data.addListenerFailure {
        Toast.makeText(this, "Error: al buscar los datos",
            Toast.LENGTH_SHORT).show()
        //Si no puede acceder a la base de datos
        se muestra mensaje de error pero de
        conexion
    }
}
```



EL editar ocupará el mismo layout pero todo ocurrirá al presionar el botón guardar

```
binding.BtGuardarDatos.setOnClickListener {
    val patente = binding.etPatenteAuto.text.toString()
    val MarcaAuto = binding.etMarcaAuto.text.toString()
    val ModeloAuto = binding.etModeloAuto.text.toString()
    val NombreDuenio = binding.etNombreDuenio.text.toString()
    databaseReference =
        FirebaseDatabase.getInstance().getReference("Datos Taxis")
    val datosNuevosTaxi = mapOf<String, String>(
        MarcaAuto to MarcaAuto,
        ModeloAuto to ModeloAuto,
        NombreDuenio to NombreDuenio
    )
    //SE CONVIERTE EL OBJETO LLAMADO A UN ARRAY
    //ASOCIATIVO CON LOS DATOS RESCATADOS
    val data =
        databaseReference.child(patente).updateChildren(datosNuevosTaxi)

    //AQUÍ SE ACTUALIZAN ↑ Y SE REALIZAN LAS
    //ACCIONES EN CASO DE ÉXITO O FALLA
    data.addSuccessListener {
        Toast.makeText(this, "Datos editados correctamente",
            Toast.LENGTH_SHORT).show()
    }
    data.addOnFailureListener {
        Toast.makeText(this, "error", Toast.LENGTH_SHORT).show()
    }
}
}
```



Eliminar Activity

Creamos el layout correspondiente (en este caso prácticamente igual a los anteriores)

Ingrese P'tente

Buscar Patente

Nombre del dueño

Marca del auto

Modelo del auto

Patente del auto

Eliminar

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".EliminarActivity"
    android:orientation="vertical"
    android:gravity="center">

    <EditText
        android:id="@+id/EtBuscarPatenteAuto"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Ingrese Patente">

    </EditText>

    <Button
        android:id="@+id/BtBuscarDatos"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Buscar Patente"
        android:layout_marginBottom="16dp">

    </Button>
    <EditText
        android:id="@+id/etNombreDuenio"
        android:layout_width="190dp"
        android:layout_height="wrap_content"
        android:hint="Nombre del dueño"></EditText>
```

```

<EditText
    android:id="@+id/etMarcaAuto"
    android:layout_width="190dp"
    android:layout_height="wrap_content"
    android:hint="Marca del auto"></EditText>

<EditText
    android:id="@+id/etModeloAuto"
    android:layout_width="190dp"
    android:layout_height="wrap_content"
    android:hint="Modelo del auto"></EditText>

<EditText
    android:id="@+id/etPatenteAuto"
    android:layout_width="190dp"
    android:layout_height="wrap_content"
    android:hint="Patente del auto"></EditText>

<Button
    android:id="@+id/btEliminarTaxi"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Eliminar"
    android:layout_marginBottom="16dp">

</Button>

</LinearLayout>

```

En el activity es lo mismo que en los anteriores, solo que ahora seteamos nuestro botón eliminar

```

binding.btEliminarTaxi.setOnClickListener {
    val patente = binding.etPatenteAuto.text.toString()
    eliminarDatos(patente)
}

```

la eliminación sera a partir de la patente

eliminarDatos(patente)

```
private fun eliminarDatos(patente:String) {
    databaseReference =
        FirebaseDatabase.getInstance().getReference("Datos Taxi")
    val data = databaseReference.child(patente).removeValue()
    data.addOnSuccessListener {
        Toast.makeText(this, "Taxi eliminado
correctamente", Toast.LENGTH_SHORT).show()

    }
    data.addOnFailureListener {
        Toast.makeText(this, "Error en la eliminacion de la patente
$patente", Toast.LENGTH_SHORT).show()
    }
}
```