



UNIVERSIDAD
TECNOLÓGICA DE
TECAMACHALCO



CODIGO FUENTE DEL PROYECTO



Materia: Desarrollo Móvil Multiplataforma

Profesor: Elvis Colotla Carvajal

Integrantes:

Jose Cristian de Jesus Torres

Harumi Catalina Cruz Cruz

Jose Humberto Cortez Cruz

Emanuel Gallardo Bravo

14 DE ABRIL DE 2022

ATOMICROBOT76

Contenido

Métodos	2
Atributos.....	10
Variables	11
Conexión a la base de datos.....	14
Componentes	15
Excepciones	17
Pruebas unitarias.....	17

Métodos

```
btnLogin.setOnClickListener(view -> {  
    userLogin();  
});
```

Método para poder dar click en el botón login que en este caso es iniciar sesión y que posteriormente te mande a un método el cual lo explicare un poco más abajo.

```
public void userLogin(){  
    String mail = txtMail.getText().toString();  
    String password = txtPassword.getText().toString();  
  
    if (TextUtils.isEmpty(mail)){  
        txtMail.setError("Ingrese un correo");  
        txtMail.requestFocus();  
    }else if (TextUtils.isEmpty(password)){  
        Toast.makeText( context: LoginActivity.this, text: "Ingrese una contraseña", Toast.LENGTH_SHORT).show();  
        txtPassword.requestFocus();  
    }else{  
  
        mAuth.signInWithEmailAndPassword(mail, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
            @Override  
            public void onComplete(@NonNull Task<AuthResult> task) {  
                if (task.isSuccessful()){  
                    Toast.makeText( context: LoginActivity.this, text: "Bienvenid@", Toast.LENGTH_SHORT).show();  
                    startActivity(new Intent( packageContext: LoginActivity.this, DashBoard.class));  
                }else {  
                    Log.w( tag: "TAG", msg: "Error:", task.getException());  
                }  
            }  
        });  
    }  
}
```

Bueno como podemos ver acá esta nuestro método userLogin el cual nos dice que tenemos que tener un email y una contraseña para poder acceder a nuestro menú, si no contamos con esto y le damos en iniciar sesión nos mandara dos mensajes el cual nos dirá que se necesita un correo y una contraseña para ingresar, acá también entramos en la parte de base de datos ya que tenemos un método donde nos hace la comparación de que si el correo y la contraseña coincide en nuestros registros esta nos conducirá a nuestro menú en donde se encontraran nuestros cuatrimestres y posteriormente nuestros apartados para subir documentos.

```
lblRegister.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) { openRegisterActivity(); }
});
```

También contamos con un método en un texto el cual al darle click a este texto nos conducirá a otro nuevo método el cual se explicará posteriormente.

```
public void openRegisterActivity() {
    Intent intent = new Intent( packageContext: this, RegisterActivity.class);
    startActivity(intent);
} // End openRegisterActivity
```

Bueno acá esta nuestro método openRegisterActivity el cual se nos manda a traer cuando le damos en el enlace de crear nueva cuenta si no cuentas con una, y esta nos mandara a una actividad nueva en la cual nos pedirá campos para poder hacer nuestros registros y estos se guarden en nuestra base de datos.

```
btnRegister.setOnClickListener(view -> {
    createuser();
});
```

Bueno este es un método dentro de nuestra Actividad de registrar nuevos usuarios, en este caso a darle crear nuevo usuario nos mandara a traer otro método el cual se llama createuser el cual explicare posteriormente para que sepas mas acerca de este evento.

```

public void createuser(){

    String name = txtUser.getText().toString();
    String mail = txtMail.getText().toString();
    String phone = txtPhone.getText().toString();
    String password = txtPassword.getEditText().getText().toString();

    if (TextUtils.isEmpty(name)){
        txtMail.setError("Ingresa un Nombre");
        txtMail.requestFocus();
    }else if (TextUtils.isEmpty(mail)){
        txtMail.setError("Ingresa un Correo");
        txtMail.requestFocus();
    }else if (TextUtils.isEmpty(phone)){
        txtMail.setError("Ingresa un Teléfono");
        txtMail.requestFocus();
    }else if (TextUtils.isEmpty(password)){
        txtMail.setError("Ingresa una Contraseña");
        txtMail.requestFocus();
    }else {

        mAuth.createUserWithEmailAndPassword(mail, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()){
                    userID = mAuth.getCurrentUser().getUid();
                    DocumentReference documentReference = db.collection("users").document(userID);

```

```

                    Map<String, Object> user = new HashMap<>();
                    user.put("Nombre", name);
                    user.put("Correo", mail);
                    user.put("Teléfono", phone);
                    user.put("Contraseña", password);

                    documentReference.set(user).addOnSuccessListener(new OnSuccessListener<Void>() {
                        @Override
                        public void onSuccess(Void unused) {
                            Log.d("TAG", "onSuccess: Datos registrados"+userID);
                        }
                    });
                    Toast.makeText(context, RegisterActivity.this, "Usuario Registrado", Toast.LENGTH_SHORT).show();
                    startActivity(new Intent(context, RegisterActivity.this, LoginActivity.class));
                }else {
                    Toast.makeText(context, RegisterActivity.this, "Usuario no registrado"+task.getException().getMessage(), Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}

```

Bueno como podemos ver en nuestro método createuser declaramos variables en este caso nuestro Nombre, Correo, Teléfono y nuestra contraseña y dentro de esto los cuales deben usar y posteriormente se hará referencia a nuestra base de datos y que nuestra colección se llamara Users y como tal lo que se guardara será el nombre, correo, teléfono y la contraseña y de esta manera esto se nos registrara en nuestra base de datos y al guardarse como nuevo usuario y el registro se haya hecho correctamente nos enviara al login activity en donde ya pondremos nuestros datos y al ser correctos nos dirigirá a un menú que posteriormente se explicara.

```

        lblLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { openLoginActivity(); }
        });

    } //End onCreate

    public void openLoginActivity() {
        Intent intent = new Intent( packageContext: this, LoginActivity.class);
        startActivity(intent);
    } // End openLoginActivity

```

Bueno dentro de nuestra actividad de registro también contamos con un apartado donde nos dice que si ya contamos con una cuenta nos dirija a nuestra actividad de login para poder entrar ya sin necesidad de hacer nuestro registro y listo.

```

        btnSalir.setOnClickListener(view -> {
            mAuth.signOut();
            startActivity(new Intent( packageContext: this, LoginActivity.class));
        });

```

Dentro de nuestro menú este contará con un botón de salir el cual solo hará la función de salir y regresarnos a nuestro login activity.

```

        unoc.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(), c1.class);
                startActivity(intent);
                finish();
            }
        });
        dosc.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(), c2.class);
                startActivity(intent);
                finish();
            }
        });
        tresc.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(), c3.class);
                startActivity(intent);
                finish();
            }
        });
    }

```

Bueno como tal dentro de nuestro menú habrá ventanas flotantes las cuales las cuales al desplazar nos mostraran unos botones los cuales estarán segmentados por cuatrimestres y estos nos redirigirán a las actividades de los cuatrimestres

donde nos dirigirán a nuestras materias y posteriormente en este hacer nuestra correspondiente subida de documentos y esto esta hecho para cada Actividad de cada cuatrimestre.

```
imagen0.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent(getApplicationContext(), DashBoard.class);  
        startActivity(intent);  
        finish();  
    }  
});
```

Bueno dentro de nuestras actividades de nuestro cuatrimestre donde nos muestra nuestras materias contaremos con una imagen que al darle click nos regresara a nuestro menú donde se nos despliegan todos nuestros cuatrimestres y así sucesivamente con cada cuatrimestre.

```
button1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent(getApplicationContext(), RetrievePDF.class);  
        startActivity(intent);  
        finish();  
    }  
});  
button2.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent(getApplicationContext(), RetrievePDF.class);  
        startActivity(intent);  
        finish();  
    }  
});
```

También como tenemos botones, en estos botones se encuentran los nombres de las materias de cada uno de nuestros cuatrimestres y al darle click estos nos mandaran a la actividad donde podremos subir nuestros documentos los cuales lo explicaremos posteriormente y esto se encuentra en cada una de nuestras pantallas de cada cuatrimestre.

```

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
        putPDF putPDF = uploadedPDF.get(i);
        Intent intent = new Intent(Intent.ACTION_VIEW);
        intent.setType("application/pdf");
        intent.setData(Uri.parse(putPDF.getUrl()));
        startActivity(intent);
    }
});

```

Dentro de nuestra actividad RetrievePDF tendremos un método el cual nos permitirá acceder a nuestros documentos y cómo podemos ver nos mostrara los documentos que en este caso ya hemos subido dentro de este encontraremos un apartado de subir archivos el cual lo explicaremos mas después.

```

private void retrievePDFFiles() {
    databaseReference = FirebaseDatabase.getInstance().getReference( path: "uploadPDF");
    databaseReference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            for (DataSnapshot ds : snapshot.getChildren()){
                putPDF putPDF = ds.getValue(com.example.sgiiii.putPDF.class);
                uploadedPDF.add(putPDF);
            }
            String[] uploadsName = new String[uploadedPDF.size()];

            for(int i=0; i<uploadsName.length;i++){
                uploadsName[i]=uploadedPDF.get(i).getName();
            }
            ArrayAdapter<String> arrayAdapter=new ArrayAdapter<~>(getApplicationContext(),
                android.R.layout.simple_expandable_list_item_1, uploadsName){
                @NonNull
                @Override
                public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
                    View view = super.getView(position, convertView, parent);
                    TextView textView = (TextView) view.findViewById(android.R.id.text1);
                    textView.setTextColor(Color.BLACK);
                    textView.setTextSize(20);
                    return view;
                }
            };
        }
    });
}

```

```

listView.setAdapter(arrayAdapter);
}
@Override
public void onCancelled(@NonNull DatabaseError error) {

}
});
}
}

```


Bueno como se explicaba anteriormente en este método ya se hace la referencia a nuestra base de datos y que en esta se guarde en este apartado nos regresara nuestros documentos como tal nos mostrara una lista de los documentos subidos.

```
editText.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View view) { selectPDF(); }
});

}

private void selectPDF() {

    Intent intent = new Intent();
    intent.setType("application/pdf");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(Intent.createChooser(intent, "PDF file select"), requestCode: 12);
}
```

Bueno como les comentaba en nuestra actividad podemos seleccionar subir archivo y nos mandara a otra actividad donde nuestro primer método se realiza en un editText el cual nos mandara a nuestros archivos de nuestros dispositivos en los cuales solo nos rescatara los PDF ya que es la única extensión de documentos que vamos a poder subir.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == 12 && resultCode == RESULT_OK && data != null && data.getData() != null) {
        btn.setEnabled(true);
        editText.setText (data.getDataString()
            .substring(data.getDataString().lastIndexOf( str: "/" ) + 1));

        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick (View view) { uploadPDFFileFirebase(data.getData()); }
        });
    }
}

private void uploadPDFFileFirebase(Uri data) {

    final ProgressDialog progressDialog = new ProgressDialog( context: this);
    progressDialog.setTitle("File is loading..");
    progressDialog.show();
}
```

```

StorageReference reference = storageReference.child("UploadPDF" + System.currentTimeMillis() + ".pdf");

reference.putFile(data).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
    @Override
    public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
        Task<Uri> uriTask = taskSnapshot.getStorage().getDownloadUrl();
        while (!uriTask.isComplete());
        Uri uri = uriTask.getResult();

        putPDF putPDF = new putPDF(editText.getText().toString(), uri.toString());
        databaseReference.child(databaseReference.push().getKey()).setValue(putPDF);
        Toast.makeText(context MainActivity2.this, text: "File Upload", Toast.LENGTH_LONG).show();
        progressDialog.dismiss();
    }
}).addOnProgressListener(new OnProgressListener<UploadTask.TaskSnapshot>() {
    @Override
    public void onProgress(@NonNull UploadTask.TaskSnapshot snapshot) {
        double progress=(100.0* snapshot.getBytesTransferred())/snapshot.getTotalByteCount();
        progressDialog.setMessage("File Uploaded.." +(int) progress+ "%");
    }
});
}

public void retrievePDF(View view) {
    startActivity (new Intent(getApplicationContext(),RetrievePDF.class));
}
}

```

Bueno y como les explicaba acá esta otro método el cual consiste que cuando se ya este seleccionado el archivo que quieres subir este se mande a nuestra base de datos el cual acá en este apartado ya esta haciendo referencia a ella y este esta pidiendo el nombre del archivo y como tal el id se genera automáticamente.

Atributos

```
public void userLogin(){  
    String mail = txtMail.getText().toString();  
    String password = txtPassword.getText().toString();
```

Nuestros atributos dentro de nuestro Login serian como tal un Correo y una Contraseña ya que debemos contar con un correo y una contraseña para poder ingresar a nuestra actividad principal donde se encuentren nuestros cuatrimestres.

```
public void createuser(){  
    String name = txtUser.getText().toString();  
    String mail = txtMail.getText().toString();  
    String phone = txtPhone.getText().toString();  
    String password = txtPassword.getText().toString();
```

Dentro de nuestra Register Activity nuestros atributos serán como tal un Nombre, un Correo, Teléfono y una Contraseña y como tal este será mandado a nuestra base de datos y como tal estos son necesarios para poder acceder desde nuestro Login.

Variables

```
private EditText txtMail;  
private TextInputLayout txtPassword;  
private Button btnLogin;  
private TextView lblRegister;  
  
private FirebaseAuth mAuth;
```

Bueno como tal estas son nuestras variables utilizadas dentro de nuestro Login como tal nuestras primeras cuatro son variables que están dentro de nuestro XML y la otra variable es de nuestra conexión de base de datos como tal eso ya se explicara posteriormente.

```
private EditText txtUser;  
private EditText txtMail;  
private EditText txtPhone;  
private TextInputLayout txtPassword;  
private Button btnRegister;  
private TextView lblLogin;  
  
private String userID;  
private FirebaseAuth mAuth;  
private FirebaseFirestore db;
```

Bueno como tal acá declaramos nuestras otras dos variables mas que las cuales serán la de Nombre y la de Teléfono y también como podemos observar agregaremos otras dos que tienen que ver con lo de Firebase en este caso en relación a nuestra base de datos que también posteriormente será explicado.

```

FloatingActionsMenu menor6;
FloatingActionButton unoc,dosc,tresc,cuatroc,cincoc;

FloatingActionsMenu mayora6;
FloatingActionButton sietec,ochoc,nuevec,diez;

private Button btnSalir;
private Button btn;
private FirebaseAuth mAuth;

```

Estas son las variables utilizadas en nuestro menú donde se encuentran los cuatrimestres y como tal son unos menus flotantes por eso están declarados como FloatingActionsMenu y también Button.

```

ImageView imagen0;

Button button1, button2, button3, button4, button5, button6, button7, button8;

```

Bueno y dentro de nuestras actividades de cada cuatrimestre tenemos estas variables las cuales serán para todas las actividades de cuatrimestre, pero declaradas con diferentes id ya que cada botón representa una materia diferente y por eso recibirán diferente id cada uno, es decir cada botón tiene un identificador distinto.

```
EditText editText;  
Button btn;  
  
StorageReference storageReference;  
DatabaseReference databaseReference;  
@Override
```

Dentro de nuestra actividad donde subiremos archivos utilizaremos estas variables y cómo podemos ver también hacemos referencia a nuestra base de datos es decir nuestras importaciones que serán de utilidad para el óptimo funcionamiento de nuestro programa a la hora de subir archivos.

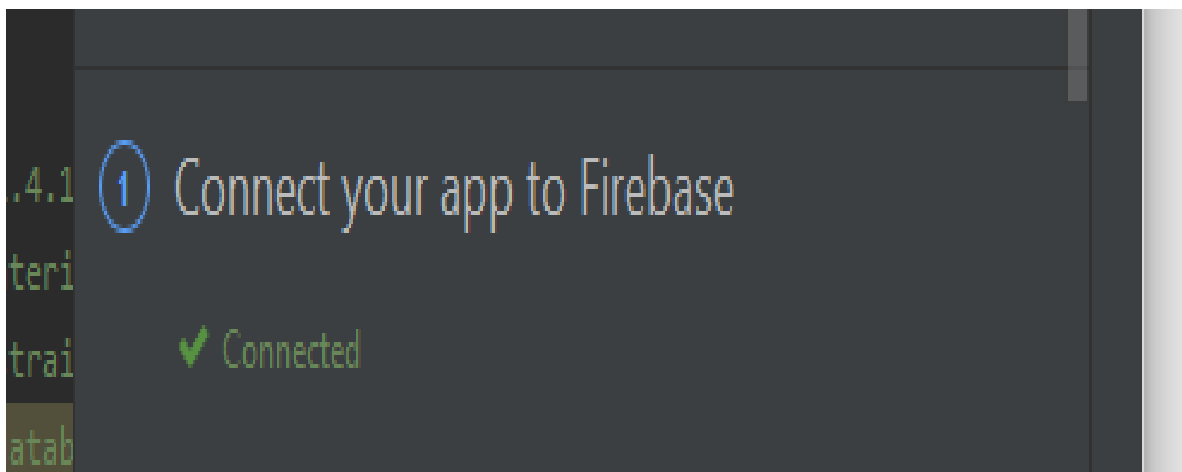
```
ListView listView;  
DatabaseReference databaseReference;  
List<putPDF> uploadedPDF;  
Button subirarchivo;
```

Bueno dentro de nuestro RetrievePDF no devolverá como tal nuestros documentos ya subidos a la base de datos y como tal estas son nuestras variables que ocuparemos dentro de esta.

Conexión a la base de datos

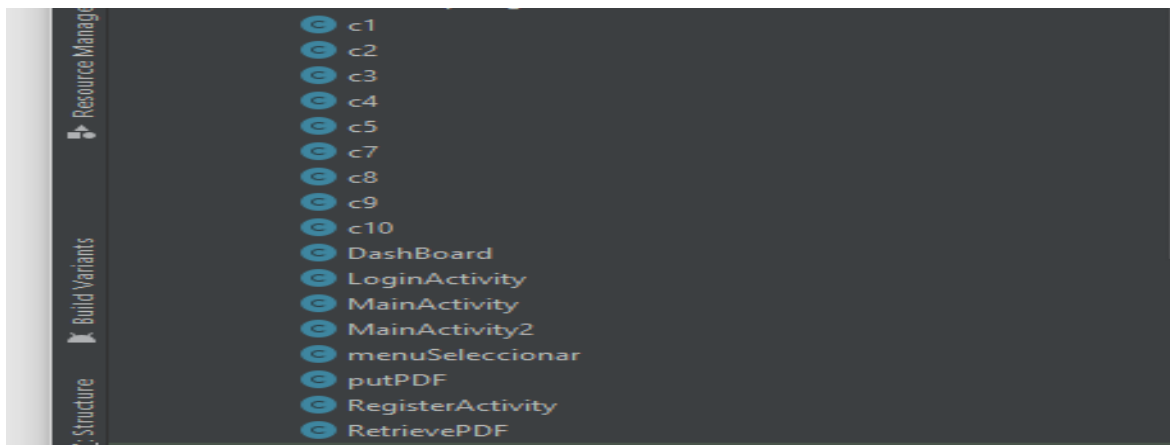
```
dependencies {  
    implementation 'androidx.appcompat:appcompat:1.4.1'  
    implementation 'com.google.android.material:material:1.5.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.1.3'  
    implementation 'com.google.firebase:firebase-database:19.4.0'  
    implementation 'com.google.firebase:firebase-storage:19.2.0'  
    implementation 'com.google.firebase:firebase-auth:19.3.2'  
    implementation 'com.google.firebase:firebase-firestore:19.0.0'  
    implementation 'com.getbase:floatingactionbutton:1.10.1'  
    testImplementation 'junit:junit:4.+'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'  
}
```

Para nosotros poder tener una buena comunicación con nuestra base de datos necesitamos hacer unas implementaciones tal y como las que se muestran un poco sombreadas con color amarillo ya que estás son necesarias para poder estar conectados con firebase y esto como tal lo hicimos desde la consola de nuestro Android studio,

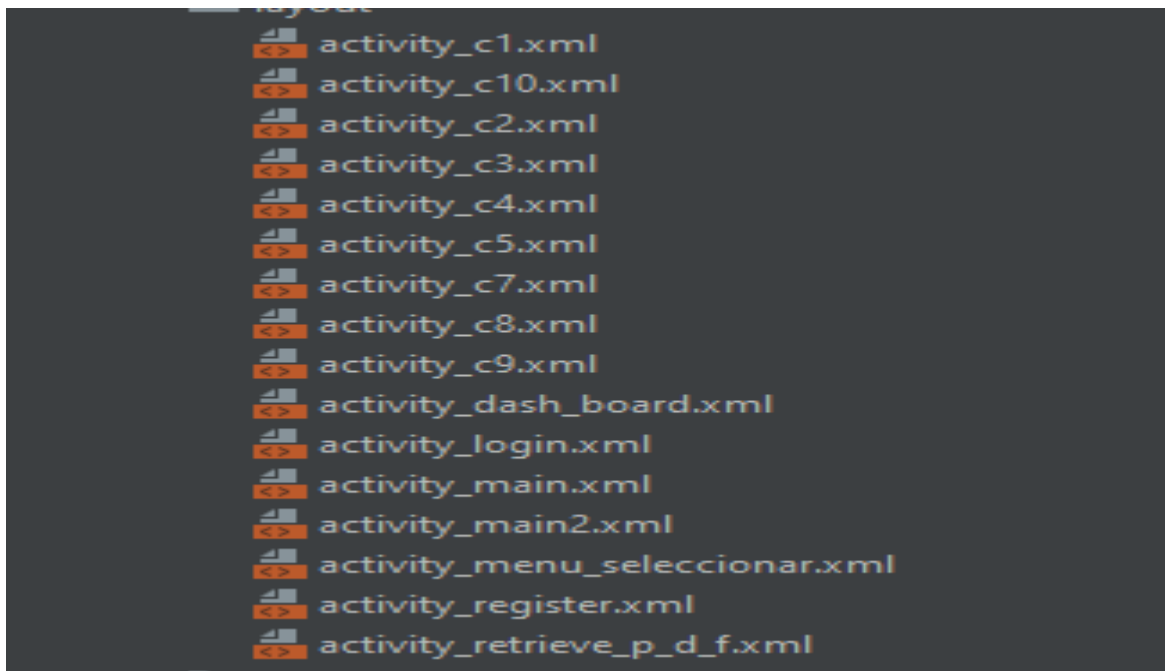


Donde nos podemos dar cuenta que nuestra base de datos esta conectada es checar en nuestro apartado de Tools y dentro de el dirigirte a Firebase y seleccionar RealtimeDatabase y se te aparecerá el apartado connected en este caso ya apareces como conectado en la base de datos y esto quiere decir que estas conectado de forma correcta.

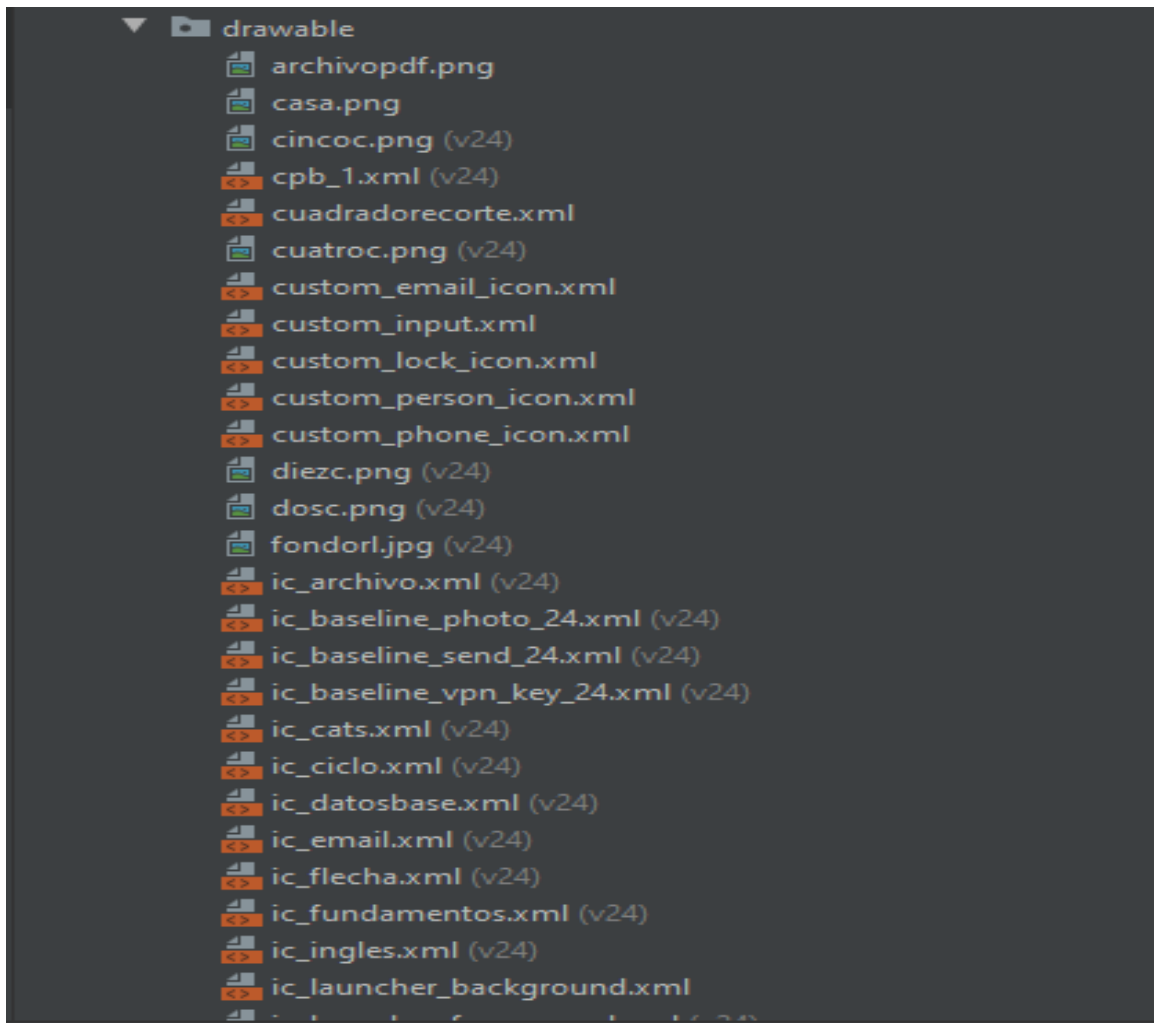
Componentes



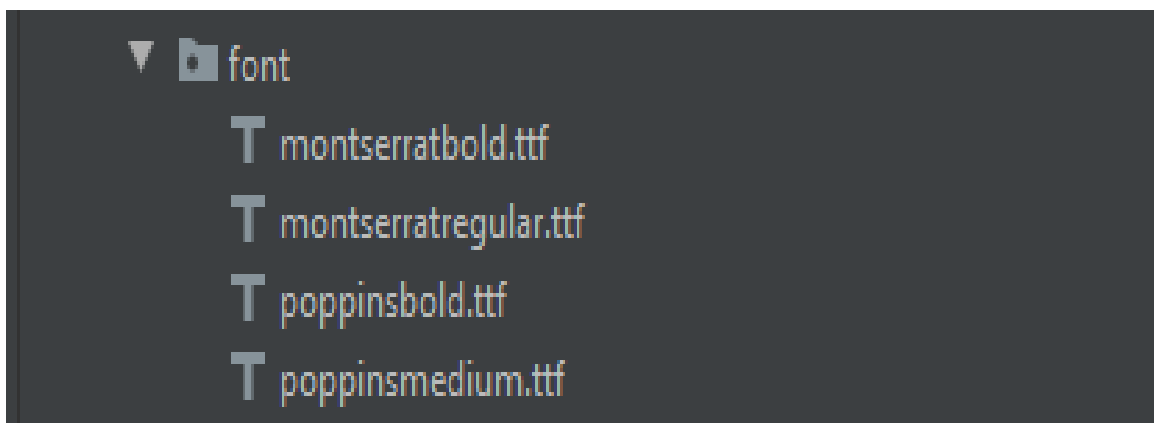
En este caso contamos con actividades las cuales ya se describieron anteriormente y que función es la que se hace dentro de ellas.



También contamos con sus respectivos Layouts en los cuales están sus diseños de cada uno de nuestras actividades y pues como lo explicó son solo los diseños de nuestras interfaces.



También contamos con una serie de imágenes que ocuparemos en nuestro proyecto es decir iconos imágenes fondos etc.

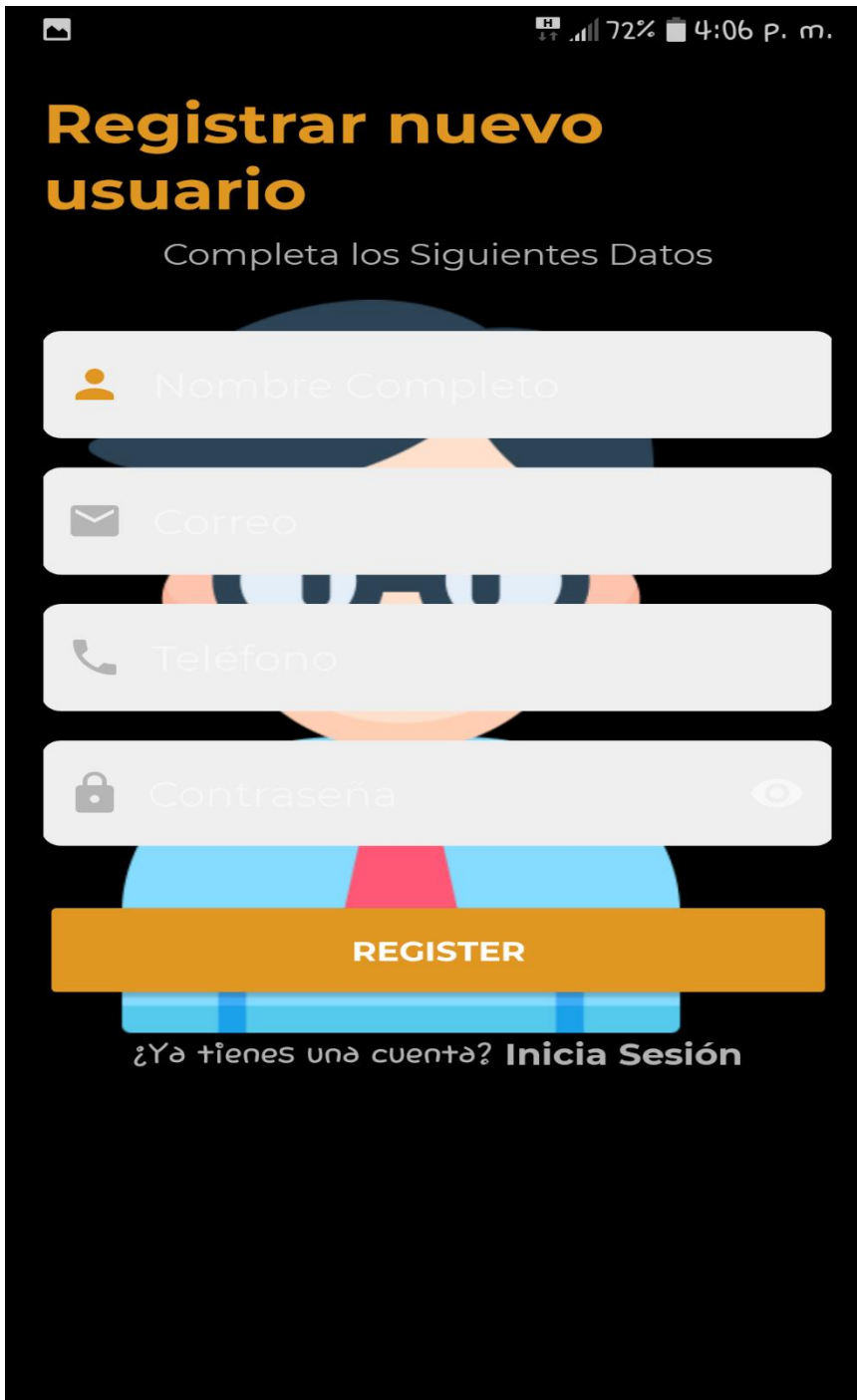


También contamos con una carpeta en la cual ya tenemos unos fondos los cuales los utilizaremos dentro de nuestros textos que aparecen en nuestra aplicación.

Excepciones

La única excepción en este caso y que nos causo ruido fue que nuestra app como tal al ejecutarla se nos detenía al hacer el intento de poder subir nuestro documento.

Pruebas unitarias



The screenshot shows a mobile application interface for registering a new user. At the top, the status bar displays a signal strength icon, a battery level of 72%, and the time 4:06 p. m. The main heading is 'Registrar nuevo usuario' in large, bold, orange letters. Below it, the instruction 'Completa los Siguients Datos' is written in a smaller, light gray font. There are four input fields, each with a light gray background and a white border. The first field is labeled 'Nombre Completo' with a person icon. The second field is labeled 'Correo' with an envelope icon. The third field is labeled 'Teléfono' with a telephone handset icon. The fourth field is labeled 'Contraseña' with a padlock icon and a toggle eye icon on the right. Below the input fields is a large, solid orange button with the word 'REGISTER' in white, uppercase letters. At the bottom, there is a link that says '¿Yà tienes una cuenta? Inicia Sesión' in white text.

Creación de un nuevo usuario de manera correcta.



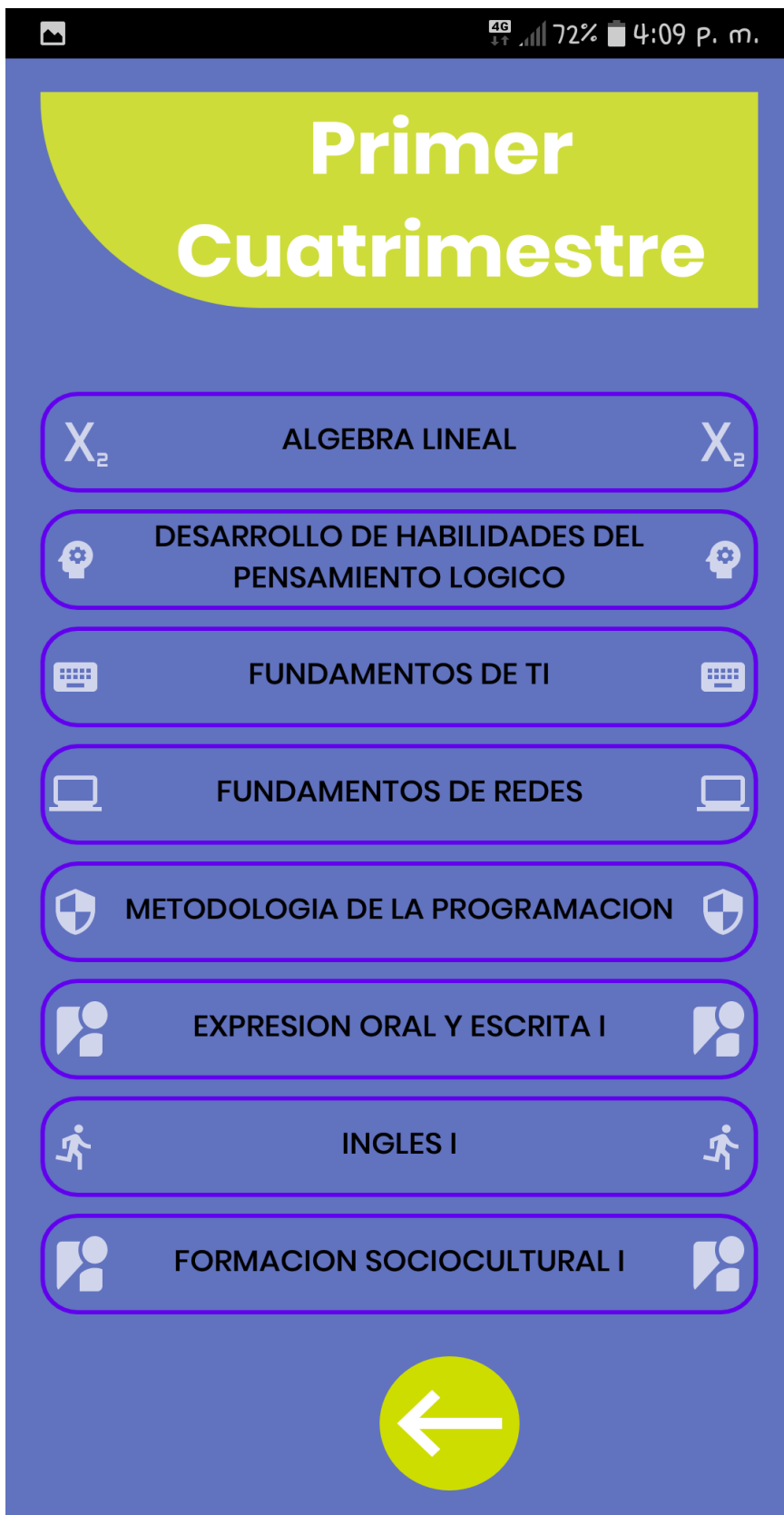
Entrar de manera correcta con un correo y una contraseña.



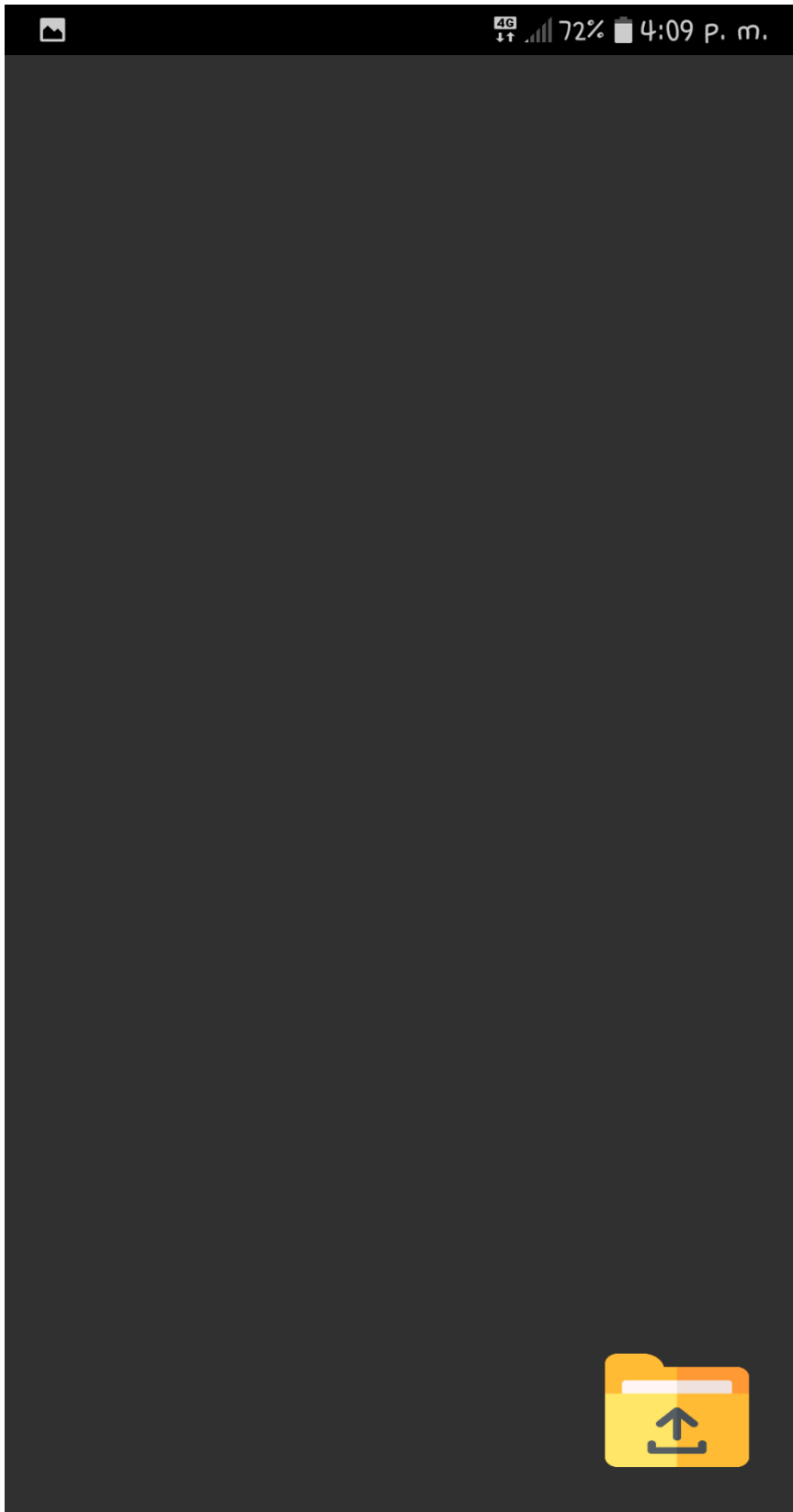
Acceso a nuestro menú correctamente para poder entrar al cuatrimestre deseado.



Al dar en el icono de más nos desplegara nuestros cuatrimestres los cuales al dar un click nos enviara a las materias y dentro de ella hacer los procesos que se piden.



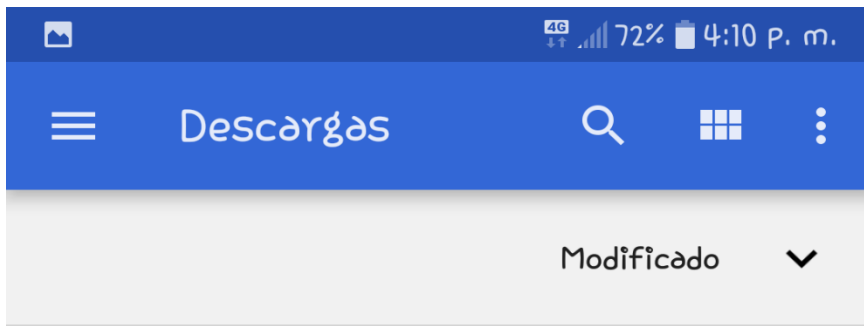
Esta es nuestra ventana de uno de nuestros cuatrimestres donde cómo podemos ver están nuestras materias y al dar click en ellas podemos ir al apartado de subir archivos.



Nos enviara a este apartado
donde al darle al icono del folder
nos mandara a la pantalla donde
por consiguiente elegiremos el
archivo y lo subiremos



Como podemos ver este es nuestro apartado para poder seleccionar y subir nuestros archivos.



pl



SNAPTUBE HACK (Actualizado) base....



#19. Como hacer menu de botones fl...



psisocpolimodal.pdf

9 abr.

1,64 MB

Documento PDF



3213-Texto del artículo-16220-1-10-2...

9 abr.

310 kB

Documento PDF



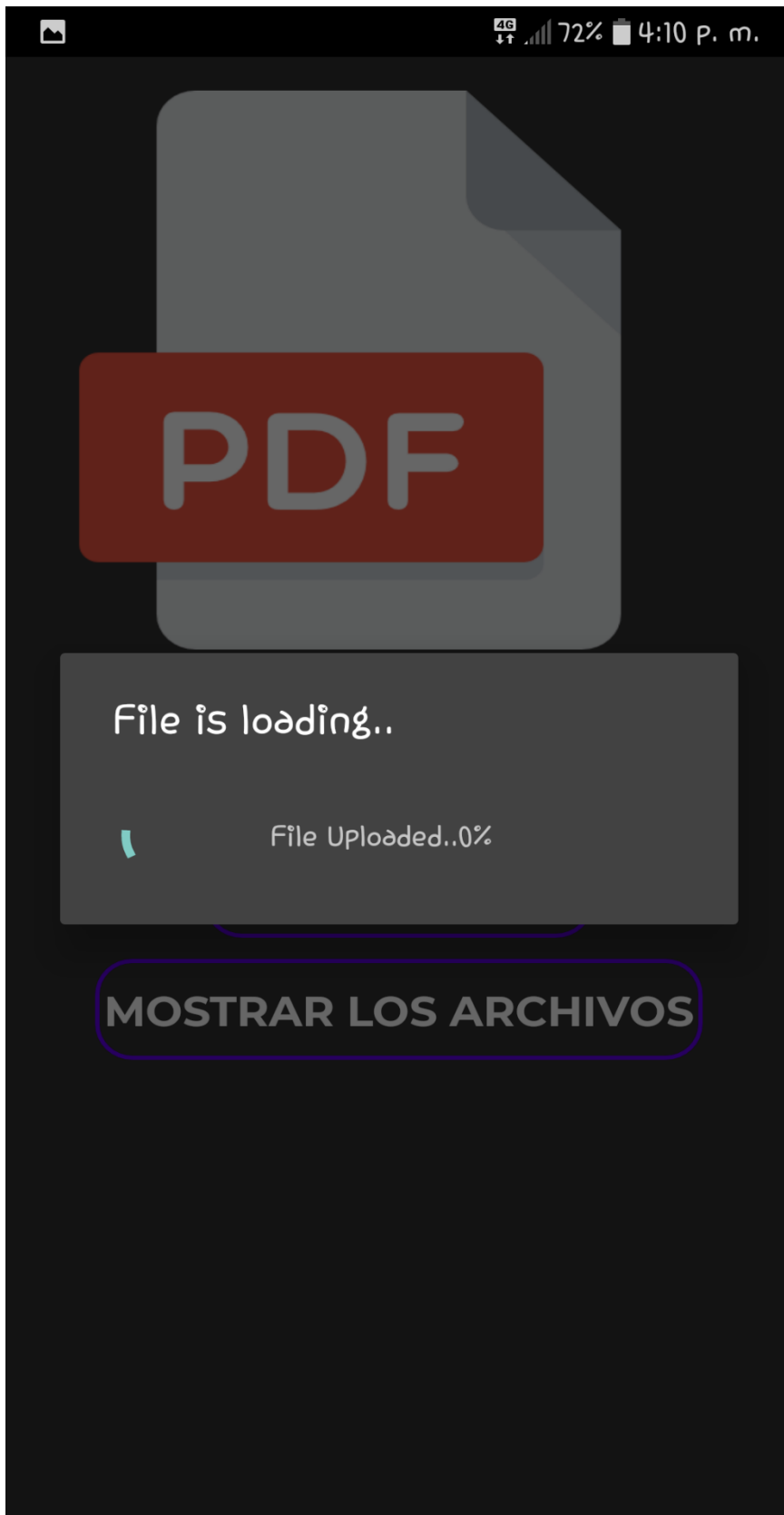
images (3).jpeg



images (2).jpeg

Al darle en el EditText nos enviara a nuestros documentos y se nos abrirá la pantalla como la que estoy mostrando a un lado.





Al seleccionar el archivo y darle en el botón subir archivo esta empezara a cargar de que el archivo se esta subiendo junto con su porcentaje de subida.