

REPORTE – PROYECTO SEGUNDO PARCIAL-PAR #5 PROGRAMACIÓN ORIENTADA A OBJETOS I PAO 2025

Grupo #3-Integrantes:

- Abraham Romero
- Danilo Ortega
- Wagner Dueñas

URL Repositorio:
https://github.com/WSmithDR/PAR5_PROY2P_Duenas_Romero_Ortega.git

Fecha: 02/09/2025

1. Presentar los botones de opciones para el usuario ingresado.
2. Creación de la sección de Ver comunicados disponibles.
3. Agregar imágenes de los comunicados en verComunicados.

Estudiante (Wagner Dueñas):

- 1. Creación de la lógica para ver mis comunicados.
- 2. Soporte para idioma ingles
- 3. Creación del Comunicado Detail.
- 4. Creación de la lógica para publicar comunicado.

3. Evidencias de Tareas

Aquí podemos observar las tareas que fueron asignadas a cada estudiante y se mostrara como evidencia los commit de esas tareas por cada estudiante.

Tipo	Clave	Resumen	Estado	Comentarios	Sprint	Persona asignada
<input checked="" type="checkbox"/>	PPDRO-11	Creacion de la logica para ver mis comunicados	FINALIZADA	Añadir comentario	Tablero Sprint 1	WR Wagner Smith Duen...
<input checked="" type="checkbox"/>	PPDRO-6	Creacion de la logica para publicar comunicado	FINALIZADA	Añadir comentario	Tablero Sprint 1	WR Wagner Smith Duen...
<input checked="" type="checkbox"/>	PPDRO-2	Presentar los botones de opciones para el usuario logeado	FINALIZADA	1 comentario	Tablero Sprint 1	DC Danilo Javier Ortega ..
<input checked="" type="checkbox"/>	PPDRO-3	Hacer Activity-Login	FINALIZADA	Añadir comentario	Tablero Sprint 1	AR Abraham Romero
<input checked="" type="checkbox"/>	PPDRO-7	Creación del formulario de PublicarComunicado	FINALIZADA	Añadir comentario	Tablero Sprint 1	AR Abraham Romero
<input checked="" type="checkbox"/>	PPDRO-9	Creacion de la excepcion datosIncompletos	FINALIZADA	Añadir comentario	Tablero Sprint 1	AR Abraham Romero
<input checked="" type="checkbox"/>	PPDRO-10	Creación de la sección de Ver comunicados disponibles	FINALIZADA	Añadir comentario	Tablero Sprint 1	DC Danilo Javier Ortega ..
<input checked="" type="checkbox"/>	PPDRO-12	Arreglos visuales en los activitys	FINALIZADA	Añadir comentario	Tablero Sprint 1	AR Abraham Romero
<input checked="" type="checkbox"/>	PPDRO-13	Agregar imagenes de los comunicados en verComunicados	FINALIZADA	Añadir comentario		DC Danilo Javier Ortega C
<input checked="" type="checkbox"/>	PPDRO-14	Soporte para idioma ingles	FINALIZADA	Añadir comentario	Tablero Sprint 2	WR Wagner Smith Duenas
<input checked="" type="checkbox"/>	PPDRO-15	Creacion del Comunicado Detail	EN CURSO	Añadir comentario	Tablero Sprint 2	WR Wagner Smith Duen...

Estudiante (Abraham Romero):

Commit 1

Commit 0338299

RomeroAbraham committed 2 days ago

Arreglos visuales ene el activity main

main (#9)

1 parent b4410aa commit 0338299

Filter files...

app/src/main

java/Exceptions

CredecialesInvalidasExcept...

DatosIncompletosExceptio...

res/layout

activity_main.xml

3 files changed +8 -5 lines changed

app/src/main/java/Exceptions/CredecialesInvalidasException.java

@@ -2,6 +2,7 @@
2 package Exceptions;
3 public class CredecialesInvalidasException extends Exception {
4 public CredecialesInvalidasException(String mensaje) {
5
5 super(mensaje);
6 }
7 }
8 }

app/src/main/java/Exceptions/DatosIncompletosException.java

@@ -2,6 +2,7 @@
2

Commit 2

Commit **8e30daf** RomeroAbraham committed 2 days ago

Cambio del logo y nombre de la app

main (#13) 1 parent [82e5fce](#) commit [8e30daf](#)


23 files changed +9 -207 lines changed

Search within code

app/src/main

- ic_launcher-playstore.png
- res
 - drawable
 - ic_launcher_backgroun...
 - ic_launcher_foregroun...
 - logo.jpeg
 - mipmap-anydpi-v26
 - ic_launcher.xml
 - ic_launcher_round.xml

57 KB



Commit 3

Commit **12d3722** RomeroAbraham committed 18 hours ago

Solucionado el problema del tema oscuro

main (#18) 1 parent [d389fe3](#) commit [12d3722](#)

6 files changed +57 -23 lines changed

Search within code

app/src/main/res

- layout
 - activity_login.xml
 - activity_mis_comunicados....
 - activity_publicar_comunica...
 - activity_ver_comunicados.x...
- values-night
 - colors.xml
- values
 - colors.xml

app/src/main/res/layout/activity_login.xml

```
41 41 android:layout_marginBottom="32dp"
42 42 android:text="@string/login_title"
43 43 android:textAlignment="center"
44 - android:textColor="@attr/colorOnSurface"
44 + android:textColor="@color/texto"
45 45 android:textSize="30sp"
46 46 android:textStyle="bold" />
47 47

59 59 android:paddingTop="12dp"
60 60 android:paddingEnd="12dp"
```

Estudiante (Danilo Ortega):

Commit 1

Commit 315217a [Browse files](#)

OrtegaDanilo authored 1 hour ago Verified

Merge pull request #19 from WSmithDR/PPDRO-15-Creacion-del-Comunicado-Detail

Ppdro 15 creacion del comunicado detail

main (#19) 2 parents: [afb6f64](#) + [a5acb49](#) commit 315217a

Filter files... 16 files changed +541 -41 lines changed Search within code

app/src/main

- AndroidManifest.xml
- java
 - Enums
 - AreaComunicado.java
 - Interfaces
 - VerComunicadoDetails...
 - Models
 - Anuncio.java
 - Comunicado.java
 - Evento.java
 - Persistencia
 - ComunicadoRepositorio...

```
... @@ -1,7 +1,9 @@
1 1 <?xml version="1.0" encoding="utf-8"?>
2 2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3 3     xmlns:tools="http://schemas.android.com/tools">
4 4 +
5 5     <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
6 6 +
7 7     <application
8 8         android:allowBackup="true"
9 9         android:dataExtractionRules="@xml/data_extraction_rules"
10 10 +
11 11     android:roundIcon="@mipmap/ic_launcher_round"
12 12     android:supportRtl="true"
13 13     android:theme="@style/Theme.PARS_PROY2P_Duenas_Romero_Ortega">
14 14 +
15 15     <activity
16 16 +
```

Commit 2

Commit b306f57 [Browse files](#)

OrtegaDanilo authored yesterday Verified

Merge pull request #16 from WSmithDR/PPDRO-13-Agregar-imagenes-de-los-comunicados-en-verComunicados

Se agrego un background en ver comunicados

main (#16) 2 parents: [ef84b80](#) + [0ba9eb3](#) commit b306f57

Filter files... 2 files changed +20 -4 lines changed Search within code

app/src/main/res

- drawable
 - background2.xml
- layout
 - activity_ver_comunicados.x...

```
... @@ -1,4 +1,13 @@
1 1 + <?xml version="1.0" encoding="utf-8"?>
2 2 <selector xmlns:android="http://schemas.android.com/apk/res/android">
3 3
4 4 +     <item>
5 5 +         <shape>
6 6 +             <gradient
7 7 +                 android:startColor="#FCB773"
8 8 +                 android:centerColor="#F4EDF9"
9 9 +                 android:endColor="#FCB773"
10 10 +                 android:angle="45" />
11 11 +         </shape>
12 12 +     </item>
13 13 </selector>
```

Commit 3

Commit **c0dbb65**

OrtegaDanilo

committed 2 days ago

Reemplazado el uso de datos de prueba por la carga desde el repositorio en VerComunicadosActivity.

main (#15)

1 parent **e3bd5ef** commit **c0dbb65**

Filter files...

app/src/main/java/com/exa...
VerComunicadosActivity.java

1 file changed +2 -1 lines changed

app/src/main/java/com/example/par5_proy2p_duenas_romero_ortega/VerComunicadosActivity.java

@@ -22,6 +22,7 @@

22 22 import Models.Comunicado;

23 23 import Models.Evento;

24 24 import Models.Usuario;

25 + import Persistencia.ComunicadoRepositorio;

25 26 import Utils.DatosDePruebaComunicados;

26 27

27 28 public class VerComunicadosActivity extends AppCompatActivity {

@@ -79,7 +80,7 @@ public void onClick(View v) {

79 80

80 81 //Acceder al archivo de comunicados

Estudiante (Wagner Dueñas):

Commit 1

Commit **a5acb49**

WSmithDR

committed 2 hours ago

Conflictos resueltos

main (#19)

2 parents **48d7da8** + **afb6f64** commit **a5acb49**

Filter files...

app/src/main
java/com/example/par5_pr...
PublicarComunicadoActivi...
res
layout
activity_login.xml
activity_main.xml
activity_mis_comunicado...
activity_publicar_comunic...
activity_ver_comunicados...
resources drawable-xxhdpi

12 files changed +171 -53 lines changed

app/src/main/java/com/example/par5_proy2p_duenas_romero_ortega/PublicarComunicadoActivity.java

@@ -37,15 +37,15 @@ public class PublicarComunicadoActivity extends AppCompatActivity {

37 37

38 38 private static final int REQUEST_CODE_PICK_IMAGE = 1001;

39 39 private ImageButton btnVolver;

40 + private Spinner spUrgencia;

40 41

41 42 private RadioGroup rgTipo;

42 43 private RadioButton rbAnuncio, rbEvento;

43 44 private Spinner spArea;

44 45 private CheckBox cbEstudiantes, cbProfesores, cbAdministrativo;

45 46 private EditText etTitulo, etLugar, etDescripcion;

46 47 private Button btnFecha, btnImagen, btnPublicar, btnCancelar;

Commit 2

Commit d00df8a

WSmithDR committed yesterday

corregidas y añadidas nuevas variables de interpolacion

main (#14)

1 parent 4c257c1 commit d00df8a

Filter files...

app/src/main/res

layout

activity_main.xml

activity_publicar_comunica...

activity_ver_comunicados.x...

values-en

strings.xml

values

strings.xml

5 files changed +22 -12 lines changed

Search within code

app/src/main/res/layout/activity_main.xml

↑

@@ -29,7 +29,7 @@

29 29 android:background="@drawable/caja_borde"

30 30 android:elevation="4dp"

31 31 android:padding="12dp"

32 - android:text="Ver Comunicados"

32 + android:text="@string/ver_comunicados"

33 33 android:textSize="18dp"

34 34 app:backgroundTint="@null"

35 35 app:cornerRadius="16dp" />

↓

@@ -42,7 +42,7 @@

42 42 android:background="@drawable/caja_borde"

43 43 android:elevation="4dp"

Commit 3

Commit 8f10711

WSmithDR committed 2 days ago

implementada serializacion de comunicados

main (#10)

1 parent 89d6072 commit 8f10711

Filter files...

app/src/main/java/com/exa...

MisComunicadosActivity.java

1 file changed +31 -3 lines changed

Search within code

app/src/main/java/com/example/par5_proy2p_duenas_romero_ortega/MisComunicadosActivity.java

↑

@@ -25,6 +25,13 @@

25 25 import Persistencia.ComunicadoRepositorio;

26 26 import Persistencia.PersistenciaOrdenamiento;

27 27 import Utilis.DatosDePruebaComunicados;

28 + import java.io.File;

29 + import java.io.FileOutputStream;

30 + import java.io.ObjectOutputStream;

31 + import java.text.SimpleDateFormat;

32 + import java.util.Date;

33 + import java.util.Locale;

34 + import android.widget.Toast;

28 35

↓

4. Identificación de teoría aplicada en programación orientada a objetos.

En esta sección deberán incluir el detalle de los bloques de código de su proyecto donde identifican cada pilar de este paradigma: Abstracción, Encapsulamiento, Herencia y Polimorfismo. Además, deberán mostrar donde usan la interface Comparable y una descripción de porqué se ha decidido aplicar el usar de cada uno de estos conceptos en sus proyectos. Si no incluyen la explicación adecuada, este punto no será válido. Ejemplo:

- **Manipulación de objetos**

```
// Título
TextView titulo = new TextView( context: this);
titulo.setText(comunicado.getTitulo());
titulo.setTextSize(25);
titulo.setTextColor(Color.BLACK);
titulo.setPadding( left: 0, top: 0, right: 0, bottom: 8);
titulo.setGravity(Gravity.CENTER);

// Imagen
ImageView imagen = new ImageView( context: this);
imagen.setScaleType(ImageView.ScaleType.CENTER_CROP);
Uri uriImagen = obtenerImagenUri(comunicado.getNombreArchivoImagen());
imagen.setImageURI(uriImagen);
imagen.setAdjustViewBounds(true);

// Descripción
TextView descripcion = new TextView( context: this);
descripcion.setText(comunicado.getDescripcion());
descripcion.setTextSize(25);
descripcion.setTextColor(Color.BLACK);
```

```
if (comunicado instanceof Evento) {
    Evento evento = (Evento) comunicado;

    TextView fecha = new TextView( context: this);
    fecha.setText("Fecha: " + evento.getFecha());
    fecha.setTextSize(20);
    fecha.setTextColor(Color.BLACK);
    fecha.setPadding( left: 0, top: 8, right: 0, bottom: 0);

    comunicadoLayout.addView(fecha);
}
```


Explicación: En el código se manipulan archivos al cargar una imagen desde el almacenamiento interno del dispositivo. Esto se realiza mediante la función `obtenerImagenUri`, que recibe el nombre del archivo de imagen (obtenido del objeto comunicado) y construye una ruta a ese archivo. Luego, se crea un objeto `Uri` a partir de esa ruta. Esta URI se pasa al método `setImageURI` del `ImageView`, permitiendo mostrar la imagen almacenada localmente. Así, el archivo de imagen es accedido y visualizado directamente desde el sistema de archivos de la app.

- Manejo de Excepciones

```
public class DatosIncompletosException extends RuntimeException {
    5 usages
    public DatosIncompletosException(String message) {
        super(message);
    }
}
```

```
public class CredencialesInvalidasException extends Exception {
    2 usages
    public CredencialesInvalidasException(String mensaje) {
        super(mensaje);
    }
}
```

```
if (imagenUri != null) {
    try (InputStream is = getContentResolver().openInputStream(imagenUri)) {
        if (is != null) {
            byte[] bytes = ManejadorArchivo.readBytesFromInputStream(is);
            String destNombre = "img_" + id + "_" + imagenNombre;
            ManejadorArchivo.escribirBinario(context, this, destNombre, bytes);
            if (c instanceof Evento) {
                ((Evento) c).setNombreArchivoImagen(destNombre);
            } else if (c instanceof Anuncio) {
                ((Anuncio) c).setNombreArchivoImagen(destNombre);
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

```
private void validarCredenciales(String username, String password) {
    try {
        Usuario usuario = Usuario.autenticar(context: this, username, password);
        String welcomeMessage = "Bienvenido {usuario.getUser()}";
        Toast.makeText(context: this, welcomeMessage, Toast.LENGTH_SHORT).show();
        Intent intent = new Intent(packageContext: this, MainActivity.class);
        Usuario.logged_user_id = usuario.getId();
        startActivity(intent);
        finish();
    } catch (CredencialesInvalidasException e) {
        Toast.makeText(context: this, e.getMessage(), Toast.LENGTH_SHORT).show();
    }
}
```

```
// Publicar
btnPublicar.setOnClickListener( View v -> {
    try {
        publicar();
    } catch (DatosIncompletosException ex) {
        Toast.makeText(context: this, ex.getMessage(), Toast.LENGTH_SHORT).show();
    }
});

// Cancelar
btnCancelar.setOnClickListener( View v -> finish());
```

Explicación: En este código se implementan excepciones personalizadas para manejar errores específicos de la aplicación. `DatosIncompletosException` extiende `RuntimeException` y se lanza cuando faltan datos requeridos, mientras que `CredencialesInvalidasException` extiende `Exception` y se usa para validar errores de autenticación. Esto permite que el programa diferencie claramente entre tipos de errores y actúe en consecuencia.

- **Lectura de Archivos**

```
public static List<String> leerArchivo(Context context, String nombreArchivo) {
    List<String> lineas = new ArrayList<>();
    try {
        FileInputStream fis = context.openFileInput(nombreArchivo);
        InputStreamReader isr = new InputStreamReader(fis);
        BufferedReader br = new BufferedReader(isr)
    } {
        String linea;
        while ((linea = br.readLine()) != null) {
            lineas.add(linea);
        }
    } catch (FileNotFoundException e) {
        System.err.println("Archivo no encontrado en almacenamiento interno: " + nombreArchivo + " [FileNotFoundException]: " + e.getMessage());
    } catch (IOException e) {
        System.err.println("Error al leer el archivo de almacenamiento interno [IOException]: " + e.getMessage());
    } catch (Exception e) {
        System.err.println("Error al leer el archivo de almacenamiento interno [Exception]: " + e.getMessage());
    }
    return lineas;
}
```

Explicación: Aquí realizamos la lectura del archivo por ejemplo el de usuarios.txt para obtener la información de cada una de sus líneas.

- **Escritura de Archivos**

```
public static void escribirArchivo(Context context, String nombre, String contenido) {
    String contenidoConSaltoDeLinea = contenido + "\n";
    try (FileOutputStream fos = context.openFileOutput(nombre, Context.MODE_APPEND)) {
        fos.write(contenidoConSaltoDeLinea.getBytes());
    } catch (IOException e) {
        System.err.println("Error al escribir el archivo [IOException]: " + e.getMessage());
    } catch (Exception e) {
        System.err.println("Error al escribir el archivo [Exception]: " + e.getMessage());
    }
}
```

Explicación: Utilizamos este método para la escritura de archivos por ejemplo el de comunicados, para que al llamar este método cada comunicado se guarde en el archivo generado

- **Serialización**

```
private void guardarListaComunicados() {
    try {
        File file = new File(getFilesDir(), "comunicados_" + Usuario.logged_user_id + ".dat");
        FileOutputStream fos = new FileOutputStream(file);
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(listaComunicados);
        oos.close();
        fos.close();
        Toast.makeText(context, this, "Lista de comunicados guardada exitosamente", Toast.LENGTH_SHORT).show();
    } catch (Exception e) {
        e.printStackTrace();
        Toast.makeText(context, this, "Error al guardar la lista de comunicados", Toast.LENGTH_SHORT).show();
    }
}
```

Explicación: En este método se utiliza la serialización para guardar una lista de objetos (listaComunicados) en un archivo interno del dispositivo. Se crea un FileOutputStream apuntando a un archivo cuyo nombre depende del usuario actual, y se envuelve en un ObjectOutputStream para poder escribir objetos.

El método writeObject(listaComunicados) convierte toda la lista en una secuencia de bytes y la almacena en el archivo. Si la operación es exitosa, se muestra un Toast de confirmación; si ocurre alguna excepción, se imprime el error y se notifica al usuario con otro Toast. Esta técnica permite persistir el estado de los datos complejos entre sesiones.

- **Interfaz Comparable**

```
37 usages 2 inheritors
public abstract class Comunicado implements Serializable, Comparable<Comunicado> {

    4 usages
    private int id;

    5 usages
    private String userId;
    5 usages

    @Override
    public int compareTo(Comunicado otro) {
        return this.getTitulo().compareToIgnoreCase(otro.getTitulo());
    }

    2 usages
    public int compareTo(Comunicado otro, OrdComunicado primaryCriteria, boolean primaryAscending,
        OrdComunicado secondaryCriteria, boolean secondaryAscending) {
        int primaryResult = compareByCriteria(otro, primaryCriteria, primaryAscending);

        if (primaryResult == 0 && secondaryCriteria != null) {
            return compareByCriteria(otro, secondaryCriteria, secondaryAscending);
        }

        return primaryResult;
    }
}
```

Explicación: La interfaz Comparable sirve para decirle a Java cómo comparar dos objetos de una misma clase, para poder ordenarlos. La clase Comunicado usa esta interfaz para ordenar los comunicados por su título, sin importar si están en mayúsculas o minúsculas.

- **Controladores de eventos**

```
//Volver al activity anterior
this.btnVolver = findViewById(R.id.backButton);
btnVolver.setOnClickListener(
    new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent( packageContext: VerComunicadosActivity.this, MainActivity.class);
            startActivity(intent);
        }
    });
```

```
rgTipo.setOnCheckedChangeListener(( RadioGroup group, int checkedId) -> {
    if (checkedId == R.id.rbEvento) {
        etLugar.setVisibility(View.VISIBLE);
        btnFecha.setVisibility(View.VISIBLE);
        tvLugarLabel.setVisibility(View.VISIBLE);
        tvFechaLabel.setVisibility(View.VISIBLE);
        spUrgencia.setVisibility(View.GONE);
        LvlUrgencia.setVisibility(View.GONE);
    } else {
        etLugar.setVisibility(View.GONE);
        btnFecha.setVisibility(View.GONE);
        tvLugarLabel.setVisibility(View.GONE);
        tvFechaLabel.setVisibility(View.GONE);
        fechaSeleccionada = "";
        btnFecha.setText(R.string.seleccionar_fecha);
        spUrgencia.setVisibility(View.VISIBLE);
        LvlUrgencia.setVisibility(View.VISIBLE);
    }
});
```

Explicación: En el primer bloque, se configura un botón para que, al hacer clic, abra otra pantalla o actividad usando un Intent, que es la forma que tiene Android de cambiar entre pantallas. En el segundo bloque, se usa un listener para un grupo de opciones (RadioGroup) que detecta cuándo cambia la opción seleccionada. Según la opción que el usuario elija, se muestran u ocultan varios elementos de la interfaz (como campos de texto y etiquetas) usando `setVisibility()`, para adaptar la pantalla y que solo muestre lo relevante. Estos controladores hacen que la app responda de forma dinámica y mejore la experiencia del usuario.

- Programación dinámica de GUI

```
<ImageButton
    android:id="@+id/backButton"
    android:layout_width="52dp"
    android:layout_height="45dp"
    android:background="?attr/listChoiceBackgroundIndicator"
    android:contentDescription="Volver"
    android:src="?attr/homeAsUpIndicator"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:tint="@color/iconColor"/>
```

```
<TextView
    android:id="@+id/txtFecha"
    android:layout_width="365dp"
    android:layout_height="45dp"
    android:text="Ninguna fecha seleccionada"
    android:textAlignment="center"
    android:textColor="#000000"
    android:textSize="20sp" />

<ScrollView
    android:id="@+id/sp_verCom"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

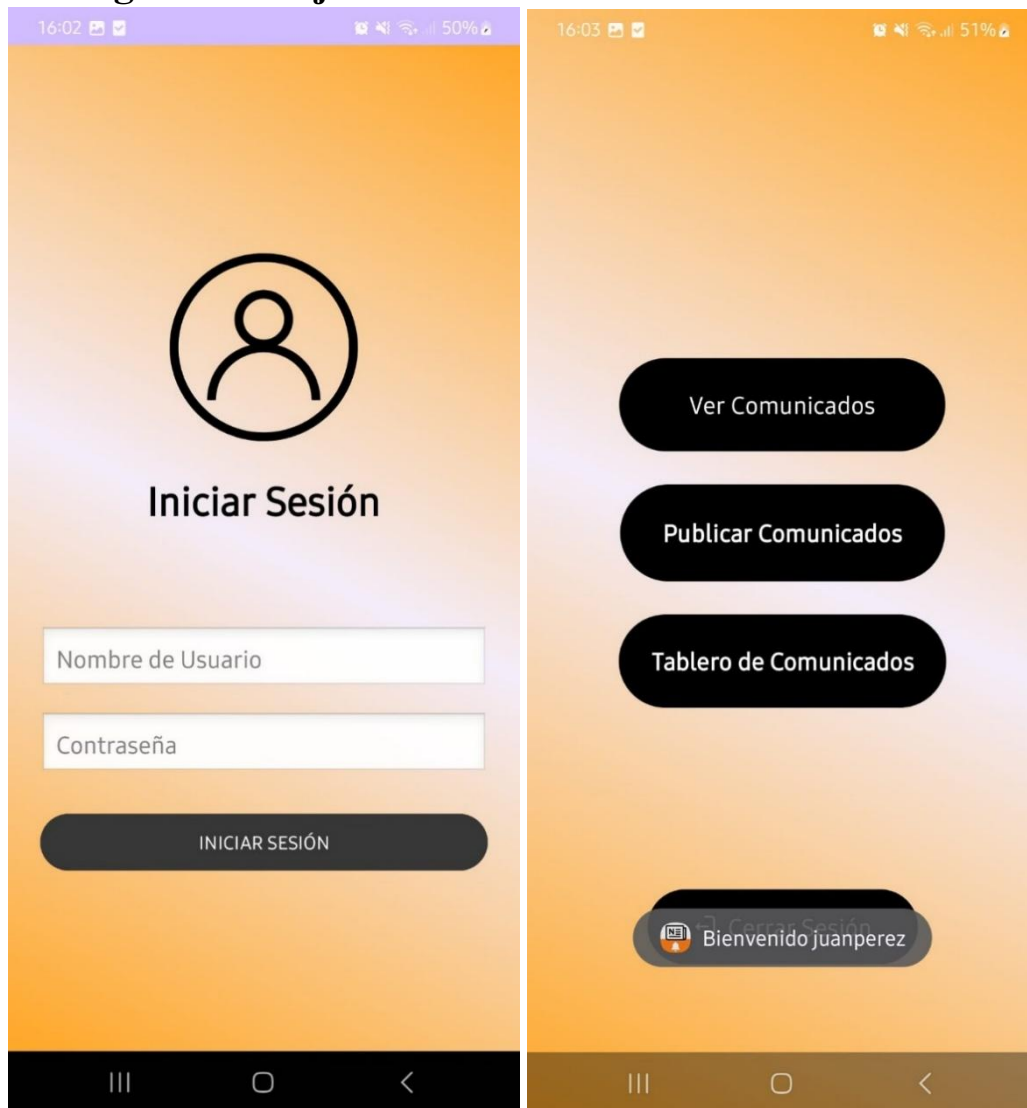
    <LinearLayout
        android:id="@+id/layout_Com"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" />

</ScrollView>
```

Explicación: Estos elementos son parte de la interfaz gráfica de la aplicación, o GUI, que es lo que el usuario ve y con lo que interactúa. El ImageButton funciona como un botón con una imagen para regresar a la pantalla anterior, con tamaño y estilo definidos para que sea claro y fácil de usar. Por otro lado, el TextView muestra texto centrado que informa al usuario, en este caso diciendo “Ninguna fecha seleccionada” como mensaje

inicial. Diseñamos estos componentes en XML para luego poder manejarlos desde el código y hacer que respondan a las acciones del usuario. Así construimos una interfaz visual funcional y amigable.

5. Programa en ejecución





16:04 52%

←

Nuevo Comunicado

Tipo:

☐ Anuncio ☐ Evento

Área:

Académico

Audiencia:

☐ Estudiantes ☐ Profesores ☐ Administrativo

Título

Ingrese el título del comunicado

Lugar:

Ingrese el lugar del comunicado

Fecha

Seleccionar fecha

Descripción:

Ingrese la descripción del comunicado

Arte (imagen):

Adjuntar imagen

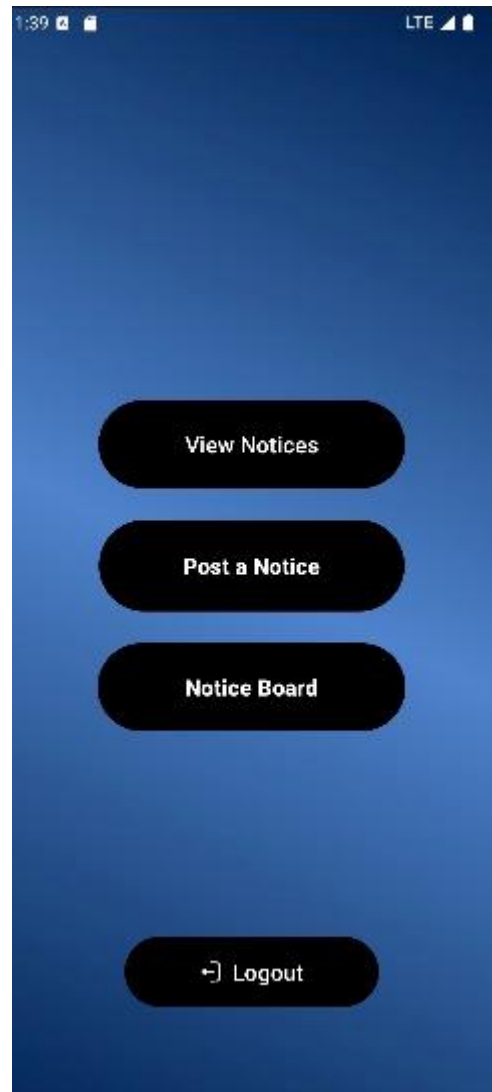
16:04 52%

←

Tablero de Comunicados

Título	Fecha
Evento1	30/09/2025

Guardar lista



JAVADOC

Agregar la documentación JAVADOC como una carpeta adicional a este reporte. Los métodos deben estar siempre comentados con el formato explicado en clase.

Referencia:<https://www.youtube.com/watch?v=KChdcRscFt0>