

Evidencia dia 4 semana 12

Android life cycle

`onCreate(Bundle)`: Se llama en la creación de la actividad (inicio), se usa para realizar todo tipo de inicializaciones, como la creación de la interfaz de usuario.

`onStart()`: Nos indica que la actividad está a punto de ser mostrada al usuario (luego de crearse en el método anterior)

`onResume()`: Se llama cuando la actividad va a comenzar a interactuar con el usuario. Es un buen lugar para lanzar las animaciones y la música.

`onPause()`: Indica que la actividad está a punto de ser lanzada a segundo plano (minimizarla), normalmente porque otra actividad es lanzada.

`onStop()`: La actividad ya no va a ser visible para el usuario.

`onRestart()`: Indica que la actividad va a volver a ser representada después de haber pasado por `onStop()`.

`onDestroy()`: Se llama antes de que la actividad sea totalmente destruida (cerrar la app). Por ejemplo, cuando el usuario pulsa el botón de volver o cuando se llama al método `finish()`.

Patrón MVP: componentes modelo vista y presentador.

CallBacks: “A” que se usa como argumento de otra función “B”

Android y Testing:

EJ: BuildVariants, permite crear diferentes versiones de la app usando el mismo código de fuente.
Permite testear la app en diferentes ambientes.

JUnit:

Conjunto de bibliotecas para hacer pruebas unitarias de aplicaciones java.

Mockito:

Probar piezas de software simulando sus dependencias externas.

Avance proyecto:

Se presenta avance de proyecto evaluación.

Creación de MainActivity Inicio que será la interfaz de log de usuario

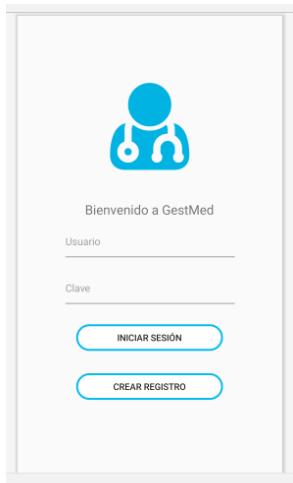
Componentes de Interfaz Inicio:

1 imageView

1 TextView

2 EditText

2 Button



Código MainActivity Inicio.

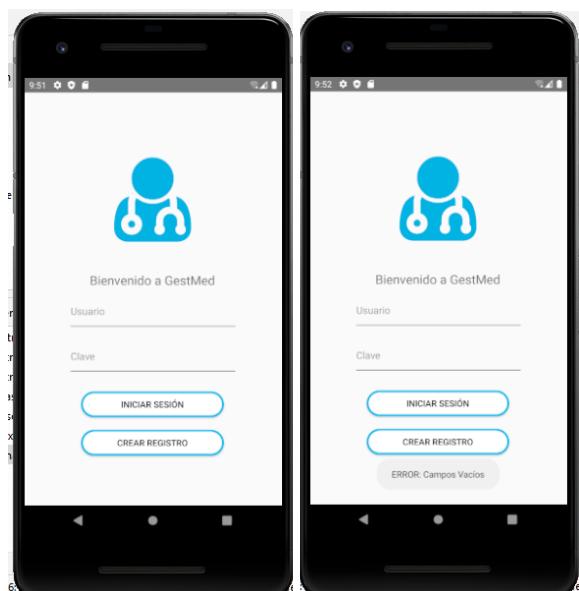
```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
    EditText user, pass;  
    Button btnEntrar, btnRegistrar;  
    DaoUsuario dao;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        user=(EditText) findViewById(R.id.User);  
        pass=(EditText) findViewById(R.id.Pass);  
        btnEntrar=(Button) findViewById(R.id.btnEntrar);  
        btnRegistrar=(Button) findViewById(R.id.btnRegistrar);  
        dao=new DaoUsuario(this);  
  
        btnEntrar.setOnClickListener(this);  
        btnRegistrar.setOnClickListener(this);  
    }  
    @Override
```

```

public void onClick(View v) {
    switch (v.getId()){
        case R.id.btnEntrar:
            String u=user.getText().toString();
            String p=pass.getText().toString();
            if(u.equals("") && p.equals ""){
                Toast.makeText(this,"ERROR: Campos Vacíos", Toast.LENGTH_LONG).show();
            }else if(dao.login(u,p)==1){
                Usuario ux= dao.getUsuario(u,p);
                Toast.makeText(this,"Sesión iniciada correctamente", Toast.LENGTH_LONG).show();
                Intent i6=new Intent(MainActivity.this, Inicio.class);
                i6.putExtra("id",ux.getId());
                startActivity(i6);
                finish();
            }
        }else{
            Toast.makeText(this,"ERROR: Usuario y/o contraseña incorrectos", Toast.LENGTH_LONG).show();
        }
    }
    break;
}
case R.id.btnRegistrar:
    Intent i=new Intent(MainActivity.this, ActivityRegistro.class);
    startActivity(i);
    break;
}
}

```

Emulador



Reflexión: Clase con bastante teoría que profundizar, avance de proyecto bien sin ninguna dificultad.