



# ITESO

Universidad Jesuita  
de Guadalajara

Tarea 3: Preferencias Compartidas

Maritza Mendoza Sicard

09/10/2017

## Preferencias Compartidas

1. Primero se creó una Actividad llamada ActivitySplashScreen, agregando también un layout en los recursos en donde puso un text view para indicar al usuario que se está cargando.

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:background="@color/colorPrimary"
    tools:context="com.iteso.dpm_s9.ActivitySplashScreen">
    <ProgressBar
        android:layout_width="100dp"
        android:layout_height="100dp" />
    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="loading..."
        android:layout_marginTop="10dp"
        android:textSize="25sp"
        android:textColor="@android:color/white"
        android:textStyle="bold"/>
</LinearLayout>
```

2. En la ActivitySplashScreen se generó un timer, porque por ahora no hay procesos externos que requieran tiempo, entonces ese estado de procesamiento se obligó para que se pueda ver la actividad. Como se puede notar en esta actividad, el proceso para saber que nueva ventana abrir es como un if y una función que se encuentra en getUser que te regresa si el usuario ha sido antiguamente loggeado o si es la primera vez que inicia. Esta función se explicará en el siguiente punto.

```
public class ActivitySplashScreen extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);

        TimerTask task = new TimerTask() {
            @Override
            public void run() {
                User user = new User().getUser(ActivitySplashScreen.this);
                Intent intent;
                if(user.isLogged()){
                    intent = new Intent(ActivitySplashScreen.this,
ActivityMain.class);
                }else{
                    intent = new Intent(ActivitySplashScreen.this,
ActivityLogin.class);
                }
                startActivity(intent);
                finish();
            }
        };

        Timer timer = new Timer();
```

```

        timer.schedule(task, 3000);
    }
}

```

3. En la clase de User se generaron los siguientes métodos:

- a) GetUser: en esta actividad se obtiene de los SharedPreferences el usuario registrado. Para esto se crea un nuevo objeto de SharedPreferences y un nuevo User, que es el valor que se va a regresar. Con el SharedPreferences, ponemos que llave queremos sacar, en este caso es el nombre del usuario, su contraseña, y si está loggeado o no.

```

public User getUser(Context context){
    SharedPreferences sharedPreferences =
context.getSharedPreferences("com.iteso.depm_s9.CACAHUATE",
    Context.MODE_PRIVATE);
    User user = new User();
    user.setUsername(sharedPreferences.getString("USER", null));
    user.setPassword(sharedPreferences.getString("PWD", null));
    user.setLogged(sharedPreferences.getBoolean("LOGGED", false));
    return user;
}

```

- b) SavePreferences: este método recibe el contexto, y con un nuevo objeto de SharedPreferences y un editor, se cambia los valores y se ingresa el nombre, la contraseña y si se pone como loggeado (se manda un true) del usuario. Finalmente se aplican los cambios.

```

public void savePreferences(Context context){
    SharedPreferences sharedPreferences =
context.getSharedPreferences("com.iteso.depm_s9.CACAHUATE",
    Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putString("USER", getUsername());
    editor.putString("PWD", getPassword());
    editor.putBoolean("LOGGED", true);
    editor.apply();
}

```

4. Cuando en el ActivitySplashScreen evalúa que el usuario está loggeado a no, decide que nueva actividad abrir. Si no lo está, te manda al ActivityLogin. Para esto se creó una nueva actividad con su respectivo layout, al cual se agregaron dos edit text para que el usuario pueda ingresar sus datos, y un botón para ingresar.

```

<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.iteso.dpm_s9.ActivityLogin">
    <LinearLayout
        android:gravity="center"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:orientation="vertical">
        <EditText
            android:id="@+id/activity_login_username"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

```

```

        android:hint="Username"
        android:inputType="textEmailAddress"/>
    <EditText
        android:id="@+id/activity_login_pwd"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"/>
</LinearLayout>
<Button
    android:id="@+id/activity_login_signin"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Login"
    android:onClick="onClick"/>
</LinearLayout>

```

5. En el ActivityLogin, se hace la función de onClick que cuando se presiona el botón, se genera un nuevo User con los datos de los edit text, y se usa el método descrito de en punto 3 de savedPreferences, guardando los nuevos datos. Finalmente te manda al ActivityMain.

```

public class ActivityLogin extends AppCompatActivity {

    protected EditText username;
    protected EditText password;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        username = (EditText) findViewById(R.id.activity_login_username);
        password = (EditText) findViewById(R.id.activity_login_pwd);
    }

    public void onClick(View view) {
        switch (view.getId()) {
            case R.id.activity_login_signin:
                User user = new User();
                user.setUsername(username.getText().toString());
                user.setPassword(password.getText().toString());
                user.savePreferences(this);
                Intent intent = new Intent(this, ActivityMain.class);
                startActivity(intent);
                finish();
                break;
        }
    }
}

```

6. Si en el ActivitySplashScreen se evalúa que el usuario ya había ingresado, entonces se va directamente al ActivityMain. En esta parte se agregó un menú donde puede hacer un logout. Eso se hizo con un layout en recursos de un menú, donde se pusieron dos ítems, uno de profile y otro de logout. Al presionar el botón de logout, se llama una función con el nombre clearPreferences, el cual elimina los datos que estaban ingresados y lo limpia. Lo hace usando un sharedPreferences y un editor. Al finalizar

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/activity_main_profile"
        android:title="Profile"/>
    <item
        android:id="@+id/activity_main_products"
        android:title="Products"/>
    <item
        android:id="@+id/activity_main_settings"
        android:title="Settings"/>
    <item
        android:id="@+id/activity_main_logout"
        android:title="LogOut" />

</menu>

```

Dentro de ActivitMain, se crea el menú y la función anteriormente descrita.

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.activity_main_menu, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.activity_main_logout:
            clearPreferences();
            return true;
        case R.id.activity_main_products:
            return true;
    }
    return true;
}

private void clearPreferences() {
    SharedPreferences sharedPreferences =
        getSharedPreferences("com.iteso.depm_s9.CACAHUATE",
            Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.clear().apply();

    Intent intent = new Intent(this, ActivityLogin.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
        Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent);
    finish();
}

```

El Proyecto se encuentra en github con el nombre de S9, el link es el siguiente:  
<https://github.com/MSicard/ProgramacionMovil>