



## AE-1. Mejora de aplicación TodoList

06/02/2022

*Jaime Aranda, Irene Alonso, Frida Abella y David Matías*

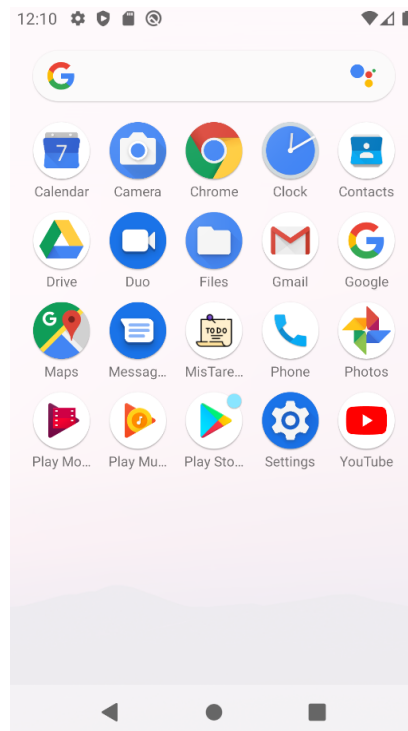
Nuestro objetivo es mejorar la aplicación que hemos realizado en la unidad 2.8, **todoList**.

GITHUB:

[https://github.com/JaimeArandaCongil/AC1\\_MOVILES](https://github.com/JaimeArandaCongil/AC1_MOVILES)

El GITHUB contiene un .ZIP con todas las carpetas del proyecto, el .apk final (la hemos probado en nuestros móviles y funciona (en principio 😊)) y este PDF de documentación.

- Cambiar el icono genérico de la app por otro de tu elección.



- Splash.



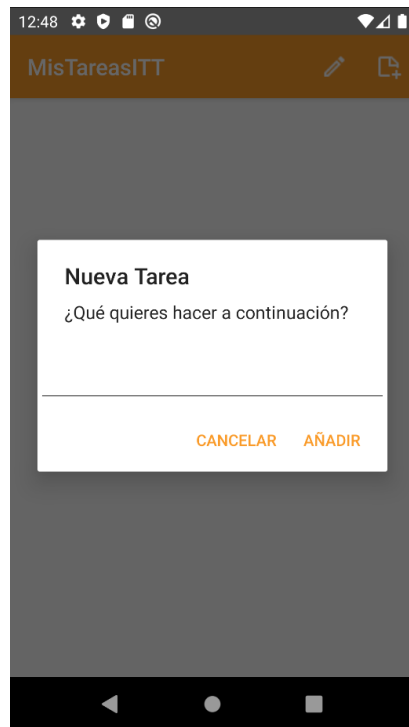
- Ventana de login.



- Crear cuentas de usuario nuevas.



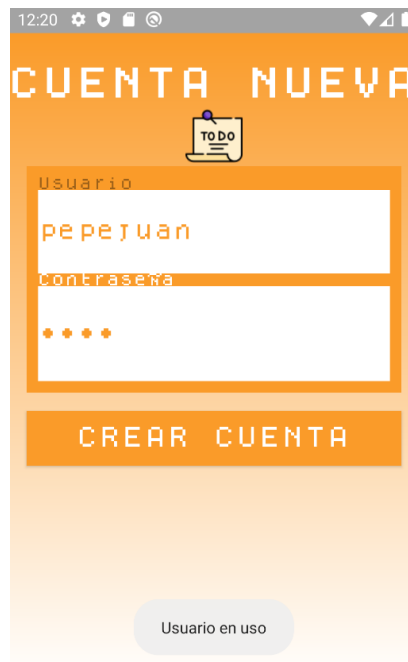
- Cada usuario verá solo sus tareas cuando se loguea y no las de los demás.
- Agregar imágenes a los botones. Cambiar el icono del botón "añadir tarea".

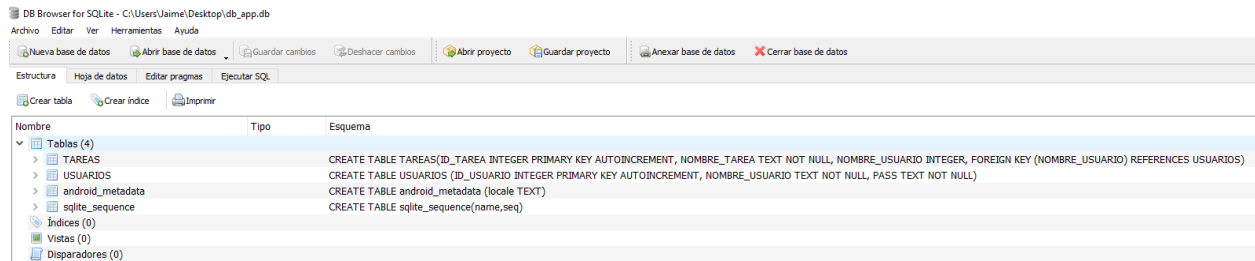


- Poner avisos del tipo Toast **personalizados** para eventos de las tareas: añadir tarea y tarea realizada.



- Hemos utilizado el programa DB SQLite Browser para controlar y comprobar que se realizan los registros en la base de datos de nuestra Aplicación.





Estructura Hoja de datos Editar pragmas Ejecutar

Tabla: **USUARIOS**

	ID_USUARIO	NOMBRE_USUARIO	PASS
	Filtro	Filtro	Filtro
1	1	hola	123
2	2		
3	3	hola1	hola1
4	4	pepejuan	1234

## ACTIVITY - CLASE → SPLASH

```
public class SplashActivity extends AppCompatActivity implements Animation.AnimationListener{

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

        getSupportActionBar().hide();

        TextView titulo = findViewById(R.id.titulo);
        Animation anim = AnimationUtils.loadAnimation(this, R.anim.animacion);
        titulo.startAnimation(anim);
        anim.setAnimationListener(this);

    }

    @Override
    public void onAnimationStart(Animation animation) {

    }

    @Override
    public void onAnimationEnd(Animation animation) {
        Intent intent = new Intent(this, LoginActivity.class);
        startActivity(intent);
        finish();

    }

    @Override
    public void onAnimationRepeat(Animation animation) {

    }

}
```

## ACTIVITY - CLASE → LOGIN

```

public class LoginActivity extends AppCompatActivity {

    ControladorDataBase controladorDB;

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

        controladorDB = new ControladorDataBase(LoginActivity.this);

        getSupportActionBar().hide();
    }

    public void login(View view){

        TextInputEditText usuario = findViewById(R.id.cajaUsuarioL);
        TextInputEditText pass = findViewById(R.id.cajaPassL);

        if(controladorDB.usuarioExiste(usuario.getText().toString())){
            Intent intent = new Intent(this, MainActivity.class);
            EditText nombreUsuario = findViewById(R.id.cajaUsuarioL);
            intent.putExtra("usuario", nombreUsuario.getText().toString());
            Toast toast = Toast.makeText(this, "Login correcto", Toast.LENGTH_LONG);
            toast.show();
            startActivity(intent);
        }else{
            Toast toast = Toast.makeText(this, "Credenciales Inválidas", Toast.LENGTH_LONG);
            toast.show();
        }
    }

    public void registerActivity(View view){

        Intent intent = new Intent(this, RegisterActivity.class);
        startActivity(intent);
    }
}

```

## ACTIVITY - CLASE → REGISTRO

```

public class RegisterActivity extends AppCompatActivity {

    ControladorDataBase controladorDB;

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

        controladorDB = new ControladorDataBase(RegisterActivity.this);

        getSupportActionBar().hide();
    }

    public void comprobarExiste(View view){

        TextInputEditText usuario = findViewById(R.id.cajaUsuarioC);
        TextInputEditText pass = findViewById(R.id.cajaPassC);

        if(controladorDB.usuarioExiste(usuario.getText().toString())){

            Toast toast = Toast.makeText(this, "Usuario en uso", Toast.LENGTH_LONG);
            toast.show();

        }else{
            controladorDB.addUsuario(usuario.getText().toString(), pass.getText().toString());
        }
    }
}

```

```

        Intent intent = new Intent(this, MainActivity.class);
        EditText nombreUsuario = findViewById(R.id.cajaUsuarioC);
        intent.putExtra("usuario", nombreUsuario.getText().toString());
        Toast toast = Toast.makeText(this, "Usuario creado", Toast.LENGTH_LONG);
        toast.show();
        startActivity(intent);
    }
}
}

```

## ACTIVITY - CLASE → CONTROLADOR BASE DE DATOS

```

public class ControladorDataBase extends SQLiteOpenHelper {

    private static final String DB_NAME = "db_app.db";
    private static final String DB_TABLE_USUARIOS = "USUARIOS";
    private static final String DB_TABLE_TAREAS = "TAREAS";
    private static final int DB_VERSION = 3;

    public ControladorDataBase(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

        db.execSQL("CREATE TABLE " + DB_TABLE_USUARIOS + " (" +
            "ID_USUARIO INTEGER PRIMARY KEY AUTOINCREMENT, " +
            "NOMBRE_USUARIO TEXT NOT NULL, " +
            "PASS TEXT NOT NULL);");

        db.execSQL("CREATE TABLE " + DB_TABLE_TAREAS + "(" +
            "ID_TAREA INTEGER PRIMARY KEY AUTOINCREMENT, " +
            "NOMBRE_TAREA TEXT NOT NULL, " +
            "NOMBRE_USUARIO INTEGER, " +
            "FOREIGN KEY (NOMBRE_USUARIO) REFERENCES USUARIOS);");

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int i, int i1) {

        db.execSQL("DROP TABLE " + DB_TABLE_USUARIOS );
        db.execSQL("DROP TABLE " + DB_TABLE_TAREAS );
        onCreate(db);

    }

    public void addTarea(String tarea, String nombre) {

        ContentValues registroUsuario = new ContentValues();
        registroUsuario.put("NOMBRE_TAREA", tarea);
        registroUsuario.put("NOMBRE_USUARIO", nombre);

        //1. ABRIR LA BD

        SQLiteDatabase db = this.getWritableDatabase();

        //2. EJECUTAR ACCIÓN
        db.insert("TAREAS", null, registroUsuario);
        //db.execSQL("INSERT INTO USUARIOS VALUES ('nombre' + 'pass' );");

        //3. CERRAR LA BD
        db.close();
    }

    public String[] obtenerTareasUsuario(String usuario) {

        SQLiteDatabase db = this.getReadableDatabase();

        Cursor cursor = db.rawQuery("SELECT * FROM TAREAS WHERE NOMBRE_USUARIO=?", new String[]{String.valueOf(usuario)});
    }
}

```

```

        int regs = cursor.getCount();
        if (regs == 0) {
            db.close();
            return null;
        } else {
            String[] tareas = new String[regs];
            cursor.moveToFirst();
            for (int i = 0; i < regs; i++) {
                tareas[i] = cursor.getString(1);
                cursor.moveToNext();
            }
            db.close();
            return tareas;
        }
    }

    public int numeroRegistros(String nombre) {

        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery("SELECT * FROM TAREAS WHERE NOMBRE_USUARIO=?", new String[]{String.valueOf(nombre)});
        return cursor.getCount();
    }

    public void borrarTarea(String tarea){
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete("TAREAS", "NOMBRE=?", new String[]{tarea});
        db.close();
    }

    public boolean usuarioExiste(String usuario){

        SQLiteDatabase db = this.getReadableDatabase();

        Cursor cursor = db.rawQuery("SELECT * FROM USUARIOS WHERE NOMBRE_USUARIO=?", new String[]{String.valueOf(usuario)} );

        int regs = cursor.getCount();
        if (regs == 0) {
            db.close();
            return false;
        } else {
            db.close();
            return true;
        }

    }

    public void addUsuario(String nombre, String pass ) {

        ContentValues registroUsuario = new ContentValues();
        registroUsuario.put("NOMBRE_USUARIO", nombre);
        registroUsuario.put("PASS", pass);

        SQLiteDatabase db = this.getWritableDatabase();
        db.insert("USUARIOS", null, registroUsuario);

        db.close();
    }
}

```

)

```

public class MainActivity extends AppCompatActivity {

    ControladorDataBase controladorDB;
    private ArrayAdapter<String> miAdapter;
    ListView listViewTareas;
    String nombre;

```



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    nombre = getIntent().getStringExtra("usuario");
    controladorDB = new ControladorDataBase(MainActivity.this);
    listViewTareas =(ListView) findViewById(R.id.lista_tareas);
    actualizarUI(nombre);

}

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

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {

    EditText cajaTexto = new EditText(this);
    AlertDialog dialog = new AlertDialog.Builder(this)
        .setTitle("Nueva Tarea")
        .setMessage("¿Qué quieres hacer a continuación?")
        .setView(cajaTexto)
        .setPositiveButton("Añadir", new DialogInterface.OnClickListener(){
            @Override
            public void onClick(DialogInterface dialogInterface, int i){
                String tarea = cajaTexto.getText().toString();
                controladorDB.addTarea(tarea, nombre);
                actualizarUI(nombre);
            }

        })
        .setNegativeButton("Cancelar", null)
        .create();
    dialog.show();

    return super.onOptionsItemSelected(item);
}

private void actualizarUI(String nombre){
    if(controladorDB.numeroRegistros(nombre)==0){
        listViewTareas.setAdapter(null);

    }else{
        miAdapter = new ArrayAdapter<>(this, R.layout.item_tarea, R.id.task_title, controladorDB.obtenerTareasUsuario(nombre));
        listViewTareas.setAdapter(miAdapter);
    }
}

public void borrarTarea(View view){

    View parent = (View) view.getParent();
    TextView tareaTextView = (TextView) parent.findViewById(R.id.task_title);
    String tarea = tareaTextView.getText().toString();
    controladorDB.borrarTarea(tarea);
    actualizarUI(nombre);

}
}

```