



# SPRINTS 4 Y 5 - Prototipo Móvil en Android Studio

---

## Introducción

En estos dos sprints, he desarrollado la aplicación móvil en Android Studio, una herramienta de gestión de asistencias, asignaturas y usuarios, mejorando el diseño original de Figma, solucionando problemas en el flujo de la aplicación y adaptándolo a Android Studio. He centrado mi desarrollo en mantener una estética coherente, funcionalidad clara y flujos de navegación intuitivos. Todo el código ha sido implementado utilizando XML para los layouts y Java para la lógica de cada pantalla.

---

## Estructura General

La aplicación cuenta con una serie de actividades que representan las diferentes áreas y funcionalidades. Cada botón y acción permite al usuario navegar de forma clara y lógica por las pantallas. El flujo principal de la aplicación es el siguiente:

1. **Pantalla principal (MainActivity):** Aquí se selecciona el rol (Alumno, Profesorado o Administrador).
  2. **Inicio de sesión (InicioActivity):** Dependiendo del rol seleccionado, se redirige al área correspondiente.
  3. **Áreas correspondientes:**
    - Área Administrador: gestión de usuarios.
    - Área Alumnado: visualización de asistencias y posibilidad de reclamarlas.
    - Área Profesorado: gestión de horarios y pasar lista.
-



## Pantalla Principal: Selección de Roles

### Código Clave

```
btnAreaAlumnado.setOnClickListener(v -> {  
    Intent intent = new Intent(MainActivity.this, InicioActivity.class);  
    intent.putExtra("ROL", "Alumno");  
    startActivity(intent);  
});
```

```
btnAreaProfesorado.setOnClickListener(v -> {  
    Intent intent = new Intent(MainActivity.this, InicioActivity.class);  
    intent.putExtra("ROL", "Profesor");  
    startActivity(intent);  
});
```

```
btnAreaAdministrador.setOnClickListener(v -> {  
    Intent intent = new Intent(MainActivity.this, InicioActivity.class);  
    intent.putExtra("ROL", "Administrador");  
    startActivity(intent);  
});
```

### Explicación

El **MainActivity** presenta botones que permiten seleccionar el área correspondiente.

Se utiliza Intent para llevar al usuario a la pantalla de inicio de sesión (**InicioActivity**) y se pasa el rol como parámetro usando intent.putExtra().


---

## Inicio de Sesión

### Mejoras Realizadas

He implementado:

- Validación de campos vacíos antes de permitir el acceso.

 <b>Institut TIC</b> de Barcelona	<b>Generalitat de Catalunya</b> <b>Departament d'Educació</b> <b>Institut TIC Barcelona</b>	<b>DAM 2º. GS PROJECTE</b> <b>SPRINTS 4 Y 5</b> Jairo Vigil Casellas
---	---	--

- Navegación a la pantalla de registro o recuperación de contraseña mediante enlaces (TextView clicables).

### Código Clave: Validación

```

btnLogin.setOnClickListener(view -> {
    String email = etEmail.getText().toString().trim();
    String password = etPassword.getText().toString().trim();

    if (email.isEmpty() || password.isEmpty()) {
        Toast.makeText(this, "Por favor, rellena todos los campos",
            Toast.LENGTH_SHORT).show();
    } else {
        String rol = getIntent().getStringExtra("ROL");
        Intent intent;


        switch (rol) {
            case "Alumno":
                intent = new Intent(InicioActivity.this, AlumnoActivity.class);
                break;
            case "Profesor":
                intent = new Intent(InicioActivity.this, ProfesorActivity.class);
                break;
            default:
                intent = new Intent(InicioActivity.this, AdministradorActivity.class);
                break;
        }

        startActivity(intent);
        finish();
    }
});

```

### Explicación

Validación sencilla para evitar que los campos de correo y contraseña estén vacíos.

 <b>Institut TIC</b> de Barcelona	<b>Generalitat de Catalunya</b> <b>Departament d'Educació</b> <b>Institut TIC Barcelona</b>	<b>DAM 2º. GS PROJECTE</b> <b>SPRINTS 4 Y 5</b> Jairo Vigil Casellas
---	---	--

Dependiendo del rol recibido, se redirige a la actividad correspondiente (Alumno, Profesorado, Administrador).

---

## Área Administrador

### Funcionalidad

Permite:

1. **Visualizar usuarios (alumnos, profesores, administradores).**
2. **Agregar un nuevo usuario.**
3. **Gestionar datos de un usuario específico.**

### Código Destacado: Navegación a Gestión de Usuario

```
btnGestionar.setOnClickListener(view -> {
    Intent intent = new Intent(AdministradorActivity.this,
    GestionarUsuarioActivity.class);
    startActivity(intent);
});
```

### Diseño


La pantalla sigue un diseño sencillo basado en filas que muestran el nombre del usuario, su rol y un botón "Gestionar" para acceder a la pantalla de modificación.

---

## Área Alumnado

### Funcionalidad: Registro de Asistencias

He implementado una lista de asistencias en formato dinámico con `LinearLayout`, mostrando asignaturas, fechas y estados de asistencia (Presente, Ausente, etc.).

 Institut TIC de Barcelona	Generalitat de Catalunya Departament d'Educació Institut TIC Barcelona	DAM 2º. GS PROJECTE SPRINTS 4 Y 5 Jairo Vigil Casellas
--	--	--

## Código Clave: Generación Dinámica

```
for (String[] asistencia : datosAsistencia) {
    TextView textView = new TextView(this);
    textView.setText(asistencia[0] + " | " + asistencia[1] + " | " + asistencia[2]);
    textView.setPadding(8, 8, 8, 8);

    if (asistencia[2].equals("Ausente")) {

textView.setTextColor(getResources().getColor(android.R.color.holo_red_dark
));
        } else {

textView.setTextColor(getResources().getColor(android.R.color.holo_green_d
ark));
        }

        llFilas.addView(textView);
    }
}
```

## Explicación

Se recorren los datos de asistencia y se añaden a la pantalla como **TextViews**.


Los colores se ajustan en función del estado de la asistencia (verde para presente, rojo para ausente).

---

## Área Profesorado

### Funcionalidad: Pase de Lista

El profesor puede marcar la asistencia de los alumnos usando **CheckBox** y registrar el estado con botones "**Presente**", "**Ausente**" y "**Retraso**".

 Institut TIC de Barcelona	Generalitat de Catalunya Departament d'Educació Institut TIC Barcelona	DAM 2º. GS PROJECTE SPRINTS 4 Y 5 Jairo Vigil Casellas
--	--	--

## Código Clave: Toasts Simples

```
btnPresente.setOnClickListener(v -> {
    Toast.makeText(this, "Asistencia registrada como Presente",
    Toast.LENGTH_SHORT).show();
});
btnAusente.setOnClickListener(v -> {
    Toast.makeText(this, "Asistencia registrada como Ausente",
    Toast.LENGTH_SHORT).show();
});
btnRetraso.setOnClickListener(v -> {
    Toast.makeText(this, "Asistencia registrada como Retraso",
    Toast.LENGTH_SHORT).show();
});
```

## Diseño


- He mantenido el diseño simple, con filas de alumnos, asignaturas y **CheckBox**.
- Los botones están alineados en la parte inferior de la pantalla.

## Botón "Salir"

En cada área principal, he añadido un botón "Salir" que devuelve al usuario a la pantalla principal (**MainActivity**).

### Lógica:

```
btnSalir.setOnClickListener(v -> {
    Intent intent = new Intent(CurrentActivity.this, MainActivity.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
    Intent.FLAG_ACTIVITY_CLEAR_TASK);
    startActivity(intent);
    finish();
});
```

 Institut TIC de Barcelona	Generalitat de Catalunya Departament d'Educació Institut TIC Barcelona	DAM 2º. GS PROJECTE SPRINTS 4 Y 5 Jairo Vigil Casellas
--	--	--

## Conclusión

Durante este sprint, he desarrollado la app siguiendo el diseño original de Figma, pero optimizándolo para Android Studio. He implementado:

- Navegación fluida entre pantallas mediante Intents.
- Layouts coherentes y adaptados al estilo de la app.
- Validación de campos para asegurar una experiencia de usuario funcional.
- Funcionalidades clave como **gestión de usuarios, registro de asistencias y pasar lista**, manteniéndolo simple y práctico.

Esta aplicación combina un diseño intuitivo con funcionalidades básicas pero bien implementadas, siendo un prototipo funcional y visualmente coherente.

Estaría lista para nuevas implementaciones y funcionalidades una vez se metiera la base de datos.