

# COMPTE RENDU

Programmation Mobile - TP2

Conception d'une interface graphique d'une application

3e année Cybersécurité

École Supérieure d'Informatique et du Numérique (ESIN)  
Collège d'Ingénierie & d'Architecture (CIA)

**Étudiant :** HATHOUTI Mohammed Taha

**Filière :** Cybersecurité

**Année :** 2025/2026

**Enseignants :** M. RAHHAL & M. GUERMAH

**Date :** 19 février 2026

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Environnement de travail</b>	<b>2</b>
2.1	Configuration . . . . .	2
2.2	Contrainte matérielle . . . . .	2
<b>3</b>	<b>Mise en page avec LinearLayout</b>	<b>3</b>
3.1	Interface mot de passe (MainActivity2) . . . . .	3
3.1.1	Code XML . . . . .	3
3.1.2	Points clés . . . . .	3
3.2	Interface avec calendrier (MainActivity3) . . . . .	4
3.2.1	Code XML . . . . .	4
3.2.2	Points clés . . . . .	5
<b>4</b>	<b>Mise en page avec TableLayout</b>	<b>6</b>
4.1	Interface type tableau (MainActivity4) . . . . .	6
4.1.1	Code XML . . . . .	6
4.1.2	Points clés . . . . .	8
<b>5</b>	<b>Mise en page avec RelativeLayout</b>	<b>8</b>
5.1	Interface avec positionnement relatif (MainActivity5) . . . . .	8
5.1.1	Code XML . . . . .	8
5.1.2	Points clés . . . . .	10
<b>6</b>	<b>Mise en page avec ConstraintLayout</b>	<b>11</b>
6.1	Interface avec positionnement par contraintes (MainActivity7) . . . . .	11
6.1.1	Code XML . . . . .	11
6.1.2	Points clés . . . . .	12
<b>7</b>	<b>Fichier de ressources strings.xml</b>	<b>13</b>
<b>8</b>	<b>Exercices</b>	<b>13</b>
8.1	Exercice 1 : Application d'achats de vêtements . . . . .	13
8.1.1	Code XML . . . . .	14
8.1.2	Points clés . . . . .	15
8.2	Exercice 2 : Fiche de saisie Smartphone . . . . .	16
8.2.1	Code XML . . . . .	16
8.2.2	Points clés . . . . .	19
<b>9</b>	<b>Difficultés rencontrées</b>	<b>20</b>
9.1	Problèmes rencontrés . . . . .	20
9.2	Solutions appliquées . . . . .	20
<b>10</b>	<b>Conclusion</b>	<b>20</b>

# 1 Introduction

Ce TP avait pour objectif l'apprentissage de la conception d'interfaces graphiques Android en utilisant différents types de layouts : `LinearLayout`, `TableLayout`, `RelativeLayout` et `ConstraintLayout`. L'objectif était de maîtriser les propriétés de positionnement et de dimensionnement des vues Android.

## 2 Environnement de travail

### 2.1 Configuration

- **OS** : Ubuntu 24.04
- **IDE** : Android Studio Hedgehog
- **SDK Android** : API 30-35
- **Émulateur** : AVD Pixel avec Android API 35

### 2.2 Contrainte matérielle

Durant ce TP, l'émulateur Android n'a pas pu être utilisé correctement en raison d'une mémoire RAM insuffisante sur la machine de travail, provoquant des crashes systématiques au lancement de l'AVD. Les rendus visuels ont donc été observés directement via l'éditeur graphique d'Android Studio. Ce problème sera résolu en migrant vers un second ordinateur disposant des ressources nécessaires, sur lequel un dual boot a déjà été installé. Les TPs suivants se dérouleront sans cette contrainte.

## 3 Mise en page avec LinearLayout

### 3.1 Interface mot de passe (MainActivity2)

Cette activité implémente une interface simple de saisie de mot de passe avec un bouton de validation.

#### 3.1.1 Code XML

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/main"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:orientation="horizontal"
9     tools:context=".MainActivity2">
10
11     <EditText
12         android:id="@+id/editTextTextPassword"
13         android:layout_width="0dp"
14         android:layout_height="wrap_content"
15         android:layout_weight="1"
16         android:autofillHints=""
17         android:ems="10"
18         android:hint="@string/mdp"
19         android:inputType="textPassword" />
20
21     <Button
22         android:layout_width="wrap_content"
23         android:layout_height="wrap_content"
24         android:text="@string/valider" />
25
26 </LinearLayout>
```

Listing 1 – activity\_main2.xml

#### 3.1.2 Points clés

- **Orientation** : Horizontale pour placer les éléments côte à côte
- **EditText** : Utilise `layout_weight="1"` et `layout_width="0dp"` pour occuper tout l'espace disponible
- **Button** : Utilise `wrap_content` pour avoir la taille minimale
- **InputType** : `textPassword` pour masquer le texte saisi

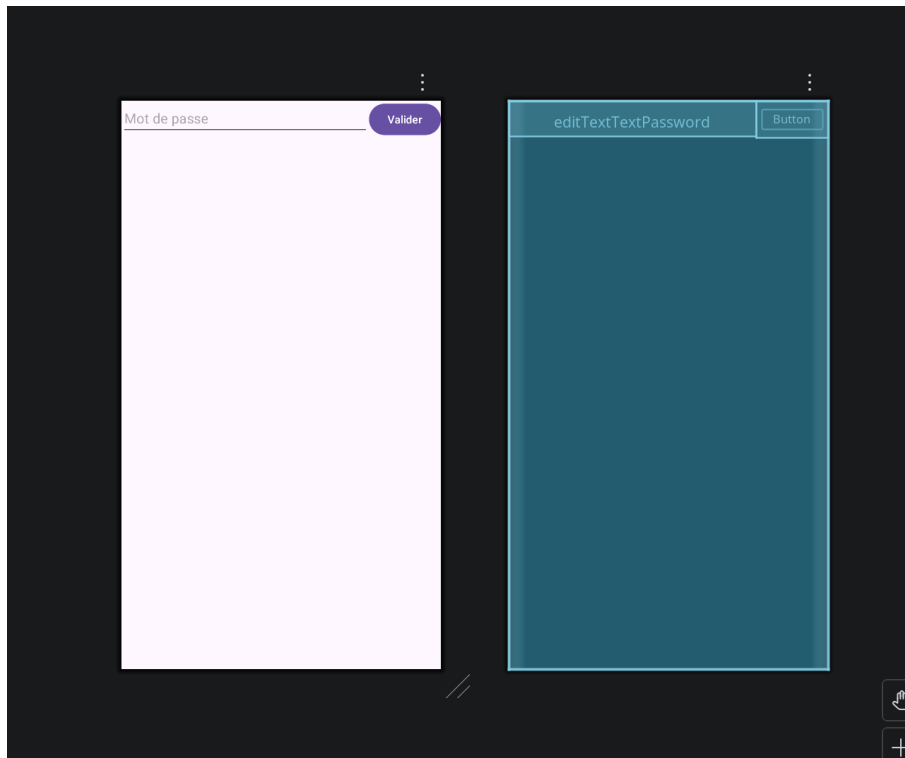


FIGURE 1 – Interface de saisie de mot de passe

## 3.2 Interface avec calendrier (MainActivity3)

Cette activité combine deux LinearLayouts : un vertical englobant et un horizontal pour la saisie.

### 3.2.1 Code XML

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/main"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:orientation="vertical"
9     tools:context=".MainActivity3">
10
11     <CalendarView
12         android:id="@+id/calendarView2"
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content" />
15
16     <LinearLayout
17         android:layout_width="match_parent"
18         android:layout_height="match_parent"
19         android:orientation="horizontal">

```

```

20
21     <EditText
22         android:id="@+id/editTextText"
23         android:layout_width="0dp"
24         android:layout_height="wrap_content"
25         android:layout_weight="1"
26         android:ems="10"
27         android:hint="@string/enregistrer_plan" />
28
29     <Button
30         android:id="@+id/button2"
31         android:layout_width="wrap_content"
32         android:layout_height="wrap_content"
33         android:text="@string/valider" />
34 </LinearLayout>
35 </LinearLayout>

```

Listing 2 – activity\_main3.xml

### 3.2.2 Points clés

- **Imbrication** : LinearLayout vertical contenant un CalendarView et un LinearLayout horizontal
- **CalendarView** : Composant natif Android pour la sélection de dates
- **Layout weight** : Même principe que la première activité pour la répartition de l'espace

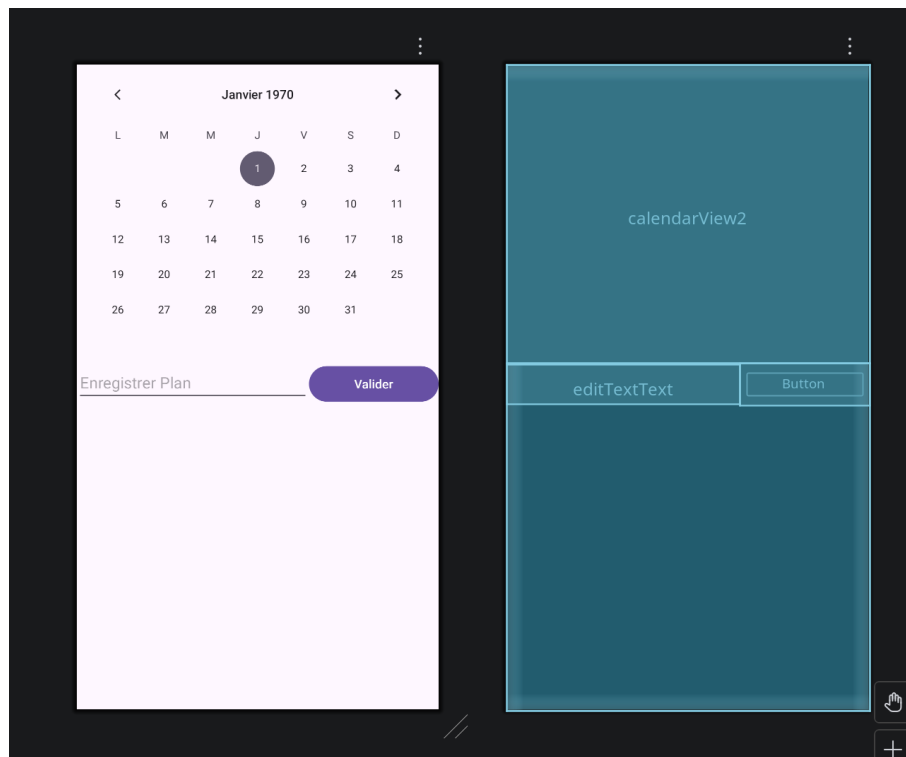


FIGURE 2 – Interface combinant calendrier et saisie

## 4 Mise en page avec TableLayout

### 4.1 Interface type tableau (MainActivity4)

Cette activité utilise TableLayout pour organiser des champs de saisie avec des boutons Valider/Annuler.

#### 4.1.1 Code XML

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <TableLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/main"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity4">
9
10    <TableRow
11        android:layout_width="match_parent"
12        android:layout_height="match_parent" >
13
14        <EditText
15            android:id="@+id/editTextText2"
16            android:layout_width="wrap_content"
17            android:layout_height="wrap_content"
18            android:ems="10"
19            android:inputType="text"
20            android:text="Messi" />
21
22        <Button
23            android:id="@+id/button3"
24            android:layout_width="wrap_content"
25            android:layout_height="wrap_content"
26            android:text="@string/valider" />
27
28        <Button
29            android:id="@+id/button"
30            android:layout_width="wrap_content"
31            android:layout_height="wrap_content"
32            android:text="@string/annuler" />
33
34    </TableRow>
35
36
37
38    <TableRow
39        android:layout_width="match_parent"
40        android:layout_height="match_parent" >
41
```

```

42     <EditText
43         android:id="@+id/editTextText30"
44         android:layout_width="wrap_content"
45         android:layout_height="wrap_content"
46         android:ems="10"
47         android:inputType="text"
48         android:text="Ronaldo" />
49
50     <Button
51         android:id="@+id/button40"
52         android:layout_width="wrap_content"
53         android:layout_height="wrap_content"
54         android:text="@string/valider" />
55
56     <Button
57         android:id="@+id/button50"
58         android:layout_width="wrap_content"
59         android:layout_height="wrap_content"
60         android:text="@string/annuler" />
61
62 </TableRow>
63
64
65 <TableRow
66     android:layout_width="match_parent"
67     android:layout_height="match_parent" >
68
69     <EditText
70         android:id="@+id/editTextText33"
71         android:layout_width="wrap_content"
72         android:layout_height="wrap_content"
73         android:ems="10"
74         android:inputType="text"
75         android:text="changement 3" />
76
77     <Button
78         android:id="@+id/button45"
79         android:layout_width="wrap_content"
80         android:layout_height="wrap_content"
81         android:text="@string/valider" />
82
83     <Button
84         android:id="@+id/button55"
85         android:layout_width="wrap_content"
86         android:layout_height="wrap_content"
87         android:text="@string/annuler" />
88 </TableRow>
89
90 </TableLayout>

```

Listing 3 – activity\_main4.xml



### 4.1.2 Points clés

- **stretchColumns** : La colonne 1 (bouton Valider) s'étire pour occuper l'espace
- **shrinkColumns** : La colonne 0 (EditText) peut se rétrécir si nécessaire
- **TableRow** : Chaque ligne est un conteneur pour les éléments de la ligne
- **Organisation** : Structure tabulaire claire avec 3 colonnes (EditText + 2 boutons)

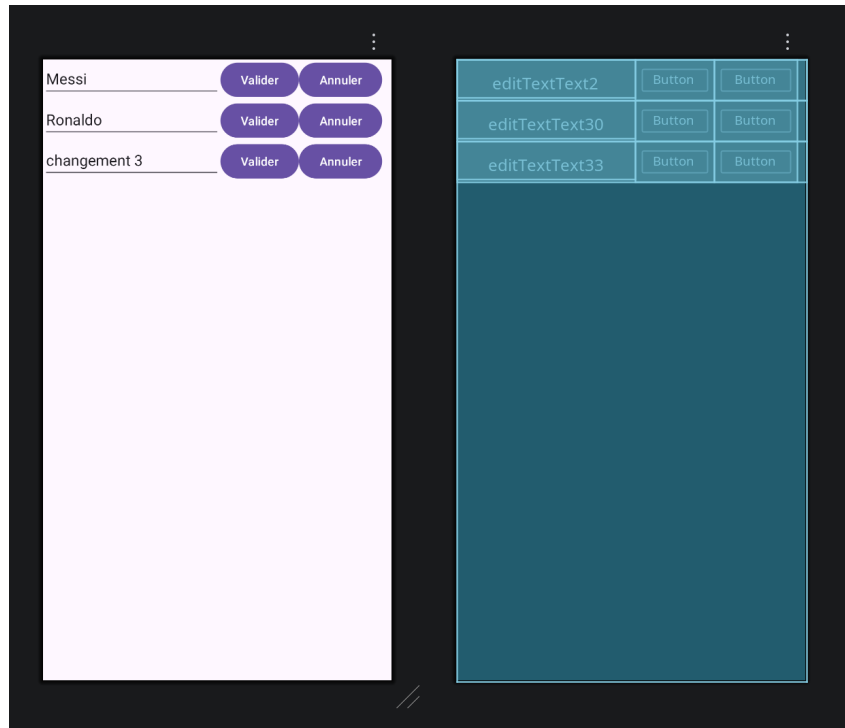


FIGURE 3 – Interface avec TableLayout

## 5 Mise en page avec RelativeLayout

### 5.1 Interface avec positionnement relatif (MainActivity5)

Cette activité utilise RelativeLayout pour positionner les boutons les uns par rapport aux autres.

#### 5.1.1 Code XML

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android=
3     "http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:id="@+id/main"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     tools:context=".MainActivity6">
```

```

10 <Button
11     android:id="@+id/btn1"
12     android:layout_width="wrap_content"
13     android:layout_height="wrap_content"
14     android:backgroundTint="#F44336"
15     android:text="@string/btn1"
16     app:layout_constraintBottom_toBottomOf="parent"
17     app:layout_constraintEnd_toEndOf="parent"
18     app:layout_constraintStart_toStartOf="parent"
19     app:layout_constraintTop_toTopOf="parent" />
20
21 <Button
22     android:id="@+id/btn2"
23     android:layout_width="wrap_content"
24     android:layout_height="wrap_content"
25     android:backgroundTint="#4CAF50"
26     android:text="@string/btn2"
27     tools:layout_editor_absoluteX="74dp"
28     tools:layout_editor_absoluteY="294dp" />
29
30 <Button
31     android:id="@+id/btn3"
32     android:layout_width="91dp"
33     android:layout_height="96dp"
34     android:backgroundTint="#00BCD4"
35     android:text="@string/btn3"
36     tools:layout_editor_absoluteX="250dp"
37     tools:layout_editor_absoluteY="294dp" />
38
39 <Button
40     android:id="@+id/btn4"
41     android:layout_width="0dp"
42     android:layout_height="wrap_content"
43     android:layout_marginStart="16dp"
44     android:layout_marginEnd="16dp"
45     android:layout_marginBottom="280dp"
46     android:backgroundTint="#E91E63"
47     android:text="@string/btn4"
48     app:layout_constraintBottom_toBottomOf="parent"
49     app:layout_constraintEnd_toEndOf="parent"
50     app:layout_constraintHorizontal_bias="0.0"
51     app:layout_constraintStart_toStartOf="parent" />
52
53 <Button
54     android:id="@+id/btn5"
55     android:layout_width="wrap_content"
56     android:layout_height="wrap_content"
57     android:layout_marginBottom="48dp"
58     android:backgroundTint="#FF9800"
59     android:text="@string/btn5"
60     app:layout_constraintBottom_toTopOf="@id/btn1"

```

```

61     app:layout_constraintStart_toStartOf="@id/btn1" />
62
63 </androidx.constraintlayout.widget.ConstraintLayout>

```

Listing 4 – activity\_main5.xml

### 5.1.2 Points clés

- **BTN1 :**
  - `constraintTop/Bottom_toTopOf/BottomOf="parent"` : Centré verticalement
  - `constraintStart/End_toStartOf/EndOf="parent"` : Centré horizontalement
- **BTN2 :**
  - `constraintEnd_toStartOf="btn1"` : À gauche de BTN1
  - `constraintTop/Bottom_toTopOf/BottomOf="btn1"` : Aligné verticalement avec BTN1
- **BTN3 :**
  - `constraintStart_toEndOf="btn1"` : À droite de BTN1
  - `constraintTop_toTopOf="btn1"` : Aligné en haut avec BTN1
- **BTN4 :**
  - `layout_width="0dp"` avec `constraintStart/End` : Occupe toute la largeur
  - `constraintBottom_toBottomOf="parent"` : Ancré en bas de l'écran
- **BTN5 :**
  - `constraintBottom_toTopOf="btn1"` : Au-dessus de BTN1
  - `constraintStart_toStartOf="btn1"` : Aligné sur le bord gauche de BTN1

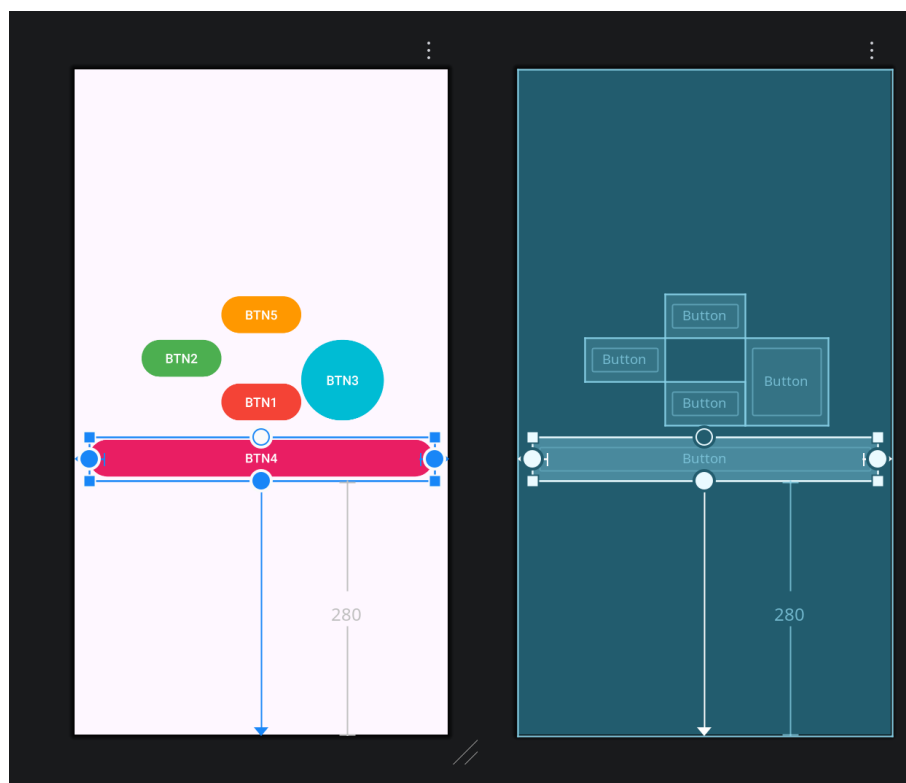


FIGURE 4 – Interface avec RelativeLayout

## 6 Mise en page avec ConstraintLayout

### 6.1 Interface avec positionnement par contraintes (MainActivity7)

Cette activité utilise ConstraintLayout pour positionner les boutons à l'aide de contraintes relatives au parent et entre eux.

#### 6.1.1 Code XML

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android=
  "http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:id="@+id/main"
6   android:layout_width="match_parent"
7   android:layout_height="match_parent"
8   tools:context=".MainActivity7">
9
10  <Button
11     android:id="@+id/btn1"
12     android:layout_width="wrap_content"
13     android:layout_height="wrap_content"
14     android:backgroundTint="#F44336"
15     android:text="@string/btn1"
16     app:layout_constraintTop_toTopOf="parent"
17     app:layout_constraintStart_toStartOf="parent"
18     app:layout_constraintEnd_toEndOf="parent"
19     app:layout_constraintVertical_bias="0.0" />
20
21  <Button
22     android:id="@+id/btn2"
23     android:layout_width="wrap_content"
24     android:layout_height="wrap_content"
25     android:backgroundTint="#00BCD4"
26     android:text="@string/btn2"
27     app:layout_constraintTop_toBottomOf="@id/btn1"
28     app:layout_constraintStart_toStartOf="parent"
29     android:layout_marginStart="16dp"
30     android:layout_marginTop="8dp" />
31
32  <Button
33     android:id="@+id/btn3"
34     android:layout_width="0dp"
35     android:layout_height="wrap_content"
36     android:backgroundTint="#4CAF50"
37     android:text="@string/btn3"
38     app:layout_constraintTop_toBottomOf="@id/btn2"
39     app:layout_constraintStart_toStartOf="parent"
40     app:layout_constraintEnd_toStartOf="@id/btn4"
```

```

41     android:layout_marginStart="8dp"
42     android:layout_marginEnd="8dp"
43     android:layout_marginTop="8dp" />
44
45     <Button
46         android:id="@+id/btn4"
47         android:layout_width="162dp"
48         android:layout_height="144dp"
49         android:backgroundTint="#942AC3"
50         android:text="@string/btn4"
51         app:layout_constraintTop_toTopOf="parent"
52         app:layout_constraintEnd_toEndOf="parent" />
53
54 </androidx.constraintlayout.widget.ConstraintLayout>

```

Listing 5 – activity\_main7.xml

### 6.1.2 Points clés

- **BTN1 :**
  - `constraintTop_toTopOf="parent"` : Collé en haut de l'écran
  - `constraintStart/End_toStartOf/EndOf="parent"` : Centré horizontalement
- **BTN2 :**
  - `constraintTop_toBottomOf="btn1"` : Positionné sous BTN1
  - `constraintStart_toStartOf="parent"` : Aligné à gauche du parent
- **BTN3 :**
  - `constraintTop_toBottomOf="btn2"` : Positionné sous BTN2
  - `layout_width="0dp"` avec `constraintStart/End` : S'étire jusqu'au bord de BTN4
- **BTN4 :**
  - `constraintTop_toTopOf="parent"` : Ancré en haut de l'écran
  - `constraintEnd_toEndOf="parent"` : Ancré à droite de l'écran
  - Dimensions fixes (162dp × 144dp) pour un rendu circulaire

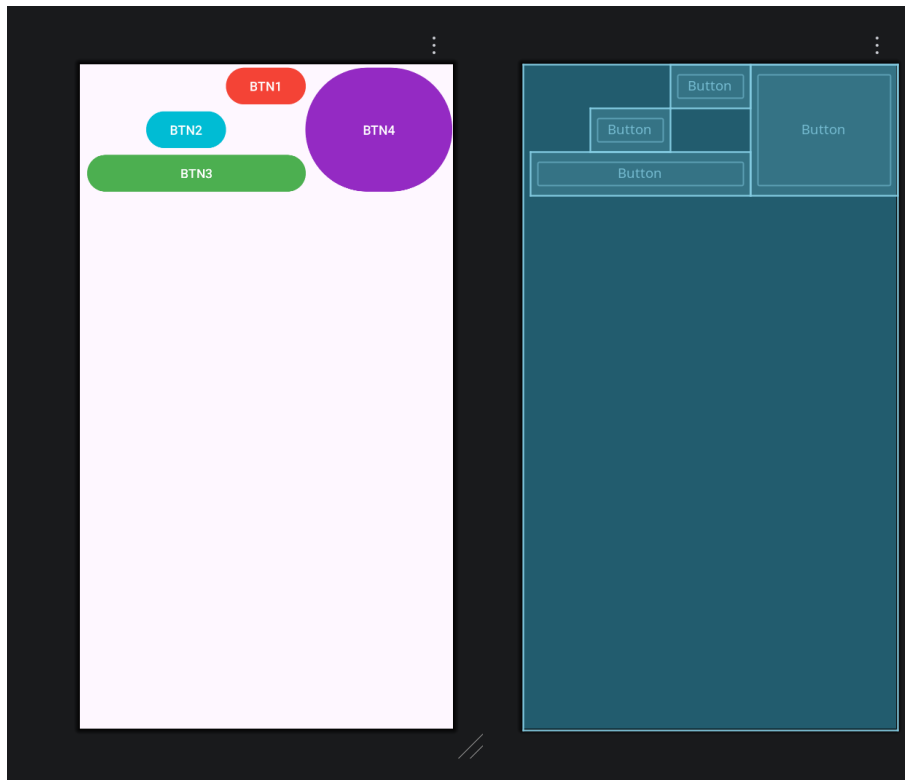


FIGURE 5 – Interface avec ConstraintLayout

## 7 Fichier de ressources strings.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <string name="app_name">TP2 Layouts</string>
4     <string name="mdp">Mot de passe</string>
5     <string name="valider">VALIDER</string>
6     <string name="annuler">ANNULER</string>
7     <string name="enregistrer_plan">Enregistrer Plan</string>
8     <string name="haut">HAUT</string>
9     <string name="droit">DROIT</string>
10    <string name="bas">BAS</string>
11 </resources>

```

Listing 6 – res/values/strings.xml

## 8 Exercices

### 8.1 Exercice 1 : Application d'achats de vêtements

Cette activité implémente un formulaire de saisie pour enregistrer des achats de vêtements lors de soldes. Elle utilise un `LinearLayout` vertical combinant plusieurs `EditText`, un `Spinner` pour le type d'article, ainsi qu'un bouton d'enregistrement.

### 8.1.1 Code XML

```
1 <resources>
2   <string name="hint_quantite">Quantité</string>
3   <string name="hint_prix">Prix (DH)</string>
4   <string name="hint_ancien_prix">Ancien prix (DH)</string>
5   <string name="hint_date">Date (JJ/MM/AAAA)</string>
6   <string name="hint_lieu">Lieu de l\'achat</string>
7   <string name="btn_enregistrer">Enregistrer</string>
8   <string name="label_type">Type d\'article</string>
9
10  <string-array name="vetements">
11    <item>Pantalon</item>
12    <item>Tricot</item>
13    <item>T-shirt</item>
14    <item>Chemise</item>
15    <item>Veste</item>
16    <item>Robe</item>
17    <item>Short</item>
18  </string-array>
19 </resources>
```

Listing 7 – strings.xml (extrait Exercice 1)

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/main"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:orientation="vertical"
8   android:padding="16dp"
9   tools:context=".Exercice1">
10
11  <TextView
12    android:layout_width="wrap_content"
13    android:layout_height="wrap_content"
14    android:text="@string/label_type"
15    android:textSize="14sp"
16    android:layout_marginBottom="4dp"/>
17
18  <Spinner
19    android:id="@+id/spinnerVetement"
20    android:layout_width="match_parent"
21    android:layout_height="wrap_content"
22    android:entries="@array/vetements"
23    android:layout_marginBottom="12dp"/>
24
25  <EditText
26    android:id="@+id/editQuantite"
27    android:layout_width="match_parent"
```

```

28     android:layout_height="wrap_content "
29     android:hint="@string/hint_quantite "
30     android:inputType="number "
31     android:layout_marginBottom="12dp"/>
32
33     <EditText
34         android:id="@+id/editPrix "
35         android:layout_width="match_parent "
36         android:layout_height="wrap_content "
37         android:hint="@string/hint_prix "
38         android:inputType="numberDecimal "
39         android:layout_marginBottom="12dp"/>
40
41     <EditText
42         android:id="@+id/editAncienPrix "
43         android:layout_width="match_parent "
44         android:layout_height="wrap_content "
45         android:hint="@string/hint_ancien_prix "
46         android:inputType="numberDecimal "
47         android:layout_marginBottom="12dp"/>
48
49     <EditText
50         android:id="@+id/editDate "
51         android:layout_width="match_parent "
52         android:layout_height="wrap_content "
53         android:hint="@string/hint_date "
54         android:inputType="date "
55         android:layout_marginBottom="12dp"/>
56
57     <EditText
58         android:id="@+id/editLieu "
59         android:layout_width="match_parent "
60         android:layout_height="wrap_content "
61         android:hint="@string/hint_lieu "
62         android:inputType="text "
63         android:layout_marginBottom="24dp"/>
64
65     <Button
66         android:id="@+id/btnEnregistrer "
67         android:layout_width="match_parent "
68         android:layout_height="wrap_content "
69         android:text="@string/btn_enregistrer"/>
70
71 </LinearLayout>

```

Listing 8 – activity\_main\_ex1.xml

### 8.1.2 Points clés

- **Spinner :**

- android:entries="@array/vetements" : Charge la liste depuis strings.xml



- Défini dans `res/values/strings.xml` via une balise `<string-array>`
- **Champs numériques :**
  - `inputType="number"` pour la quantité (entier)
  - `inputType="numberDecimal"` pour le prix et l'ancien prix (décimal)
- **Champ date :**
  - `inputType="date"` pour restreindre la saisie au format date
- **Layout :**
  - `LinearLayout` vertical avec `padding="16dp"` pour l'espacement
  - Tous les champs ont `layout_width="match_parent"` pour occuper toute la largeur

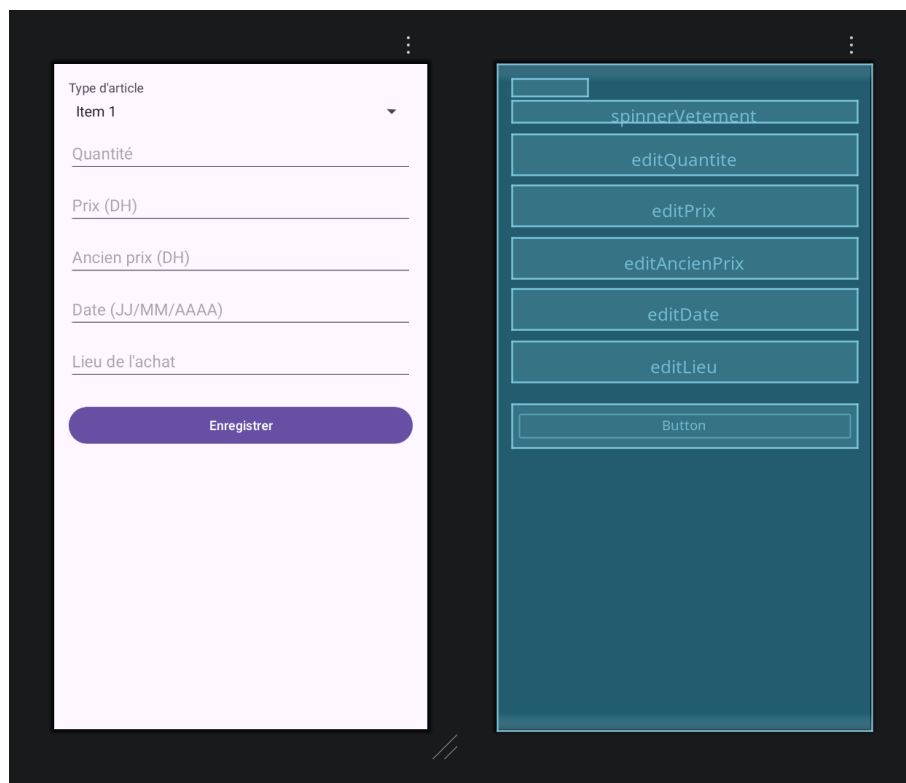


FIGURE 6 – Interface de l'application d'achats de vêtements

## 8.2 Exercice 2 : Fiche de saisie Smartphone

Cette activité implémente un formulaire de saisie pour décrire un smartphone. Elle utilise un `ScrollView` englobant un `LinearLayout` vertical, afin de gérer le défilement si le contenu dépasse la hauteur de l'écran. La notation est réalisée via un `RatingBar`.

### 8.2.1 Code XML

```

1 <resources>
2   <string name="hint_nom">Nom du smartphone</string>
3   <string name="hint_fabricant">Fabricant</string>
4   <string name="hint_annee">Année de sortie</string>

```

```

5     <string name="hint_processeur">Processeur</string>
6     <string name="hint_ram">RAM (Go)</string>
7     <string name="hint_cam_front">Caméra frontale (MP)</string>
8     <string name="hint_cam_arriere">Caméra arri re (MP)</string>
9     <string name="label_notation">Notation</string>
10    <string name="btn_sauvegarder">Sauvegarder</string>
11 </resources>

```

Listing 9 – strings.xml (extrait Exercice 2)

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <ScrollView xmlns:android="http://schemas.android.com/apk/res/
    android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:id="@+id/main"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".Exercice2">
9
10     <LinearLayout
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:orientation="vertical"
14         android:padding="16dp">
15
16         <EditText
17             android:id="@+id/editNom"
18             android:layout_width="match_parent"
19             android:layout_height="wrap_content"
20             android:hint="@string/hint_nom"
21             android:inputType="text"
22             android:layout_marginBottom="12dp"/>
23
24         <EditText
25             android:id="@+id/editFabricant"
26             android:layout_width="match_parent"
27             android:layout_height="wrap_content"
28             android:hint="@string/hint_fabricant"
29             android:inputType="text"
30             android:layout_marginBottom="12dp"/>
31
32         <EditText
33             android:id="@+id/editAnnee"
34             android:layout_width="match_parent"
35             android:layout_height="wrap_content"
36             android:hint="@string/hint_annee"
37             android:inputType="number"
38             android:layout_marginBottom="12dp"/>
39
40         <EditText
41             android:id="@+id/editProcesseur"

```

```

42         android:layout_width="match_parent"
43         android:layout_height="wrap_content"
44         android:hint="@string/hint_processeur"
45         android:inputType="text"
46         android:layout_marginBottom="12dp"/>
47
48     <EditText
49         android:id="@+id/editRam"
50         android:layout_width="match_parent"
51         android:layout_height="wrap_content"
52         android:hint="@string/hint_ram"
53         android:inputType="number"
54         android:layout_marginBottom="12dp"/>
55
56     <EditText
57         android:id="@+id/editCamFront"
58         android:layout_width="match_parent"
59         android:layout_height="wrap_content"
60         android:hint="@string/hint_cam_front"
61         android:inputType="number"
62         android:layout_marginBottom="12dp"/>
63
64     <EditText
65         android:id="@+id/editCamArriere"
66         android:layout_width="match_parent"
67         android:layout_height="wrap_content"
68         android:hint="@string/hint_cam_arriere"
69         android:inputType="number"
70         android:layout_marginBottom="12dp"/>
71
72     <TextView
73         android:layout_width="wrap_content"
74         android:layout_height="wrap_content"
75         android:text="@string/label_notation"
76         android:textSize="14sp"
77         android:layout_marginBottom="4dp"/>
78
79     <RatingBar
80         android:id="@+id/ratingBar"
81         android:layout_width="wrap_content"
82         android:layout_height="wrap_content"
83         android:numStars="5"
84         android:stepSize="0.5"
85         android:rating="0"
86         android:layout_marginBottom="24dp"/>
87
88     <Button
89         android:id="@+id/btnSauvegarder"
90         android:layout_width="match_parent"
91         android:layout_height="wrap_content"
92         android:text="@string/btn_sauvegarder"/>

```

```

93         </LinearLayout>
94     </ScrollView>
95

```

Listing 10 – activity\_main\_ex2.xml

### 8.2.2 Points clés

- **ScrollView :**
  - Enveloppe le `LinearLayout` pour permettre le défilement vertical
  - Le `LinearLayout` enfant a `layout_height="wrap_content"`
- **Champs numériques :**
  - `inputType="number"` pour l'année, la RAM et les résolutions caméra
  - Empêche la saisie de caractères non numériques
- **RatingBar :**
  - `android:numStars="5"` : fixe le nombre d'étoiles à 5
  - `android:stepSize="0.5"` : granularité de 0,5 étoile
  - `layout_width="wrap_content"` : taille adaptée au nombre d'étoiles

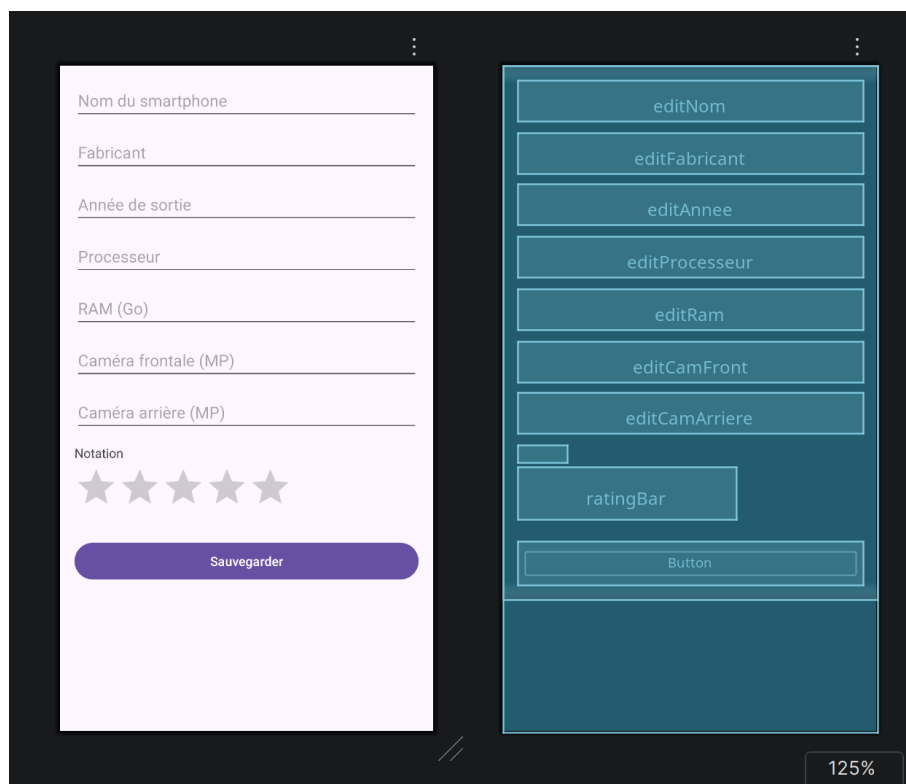


FIGURE 7 – Interface de la fiche de saisie Smartphone

## 9 Difficultés rencontrées

### 9.1 Problèmes rencontrés

- **Émulateur insuffisant en RAM** : L'émulateur Android crashait systématiquement au lancement en raison de ressources mémoire insuffisantes sur la machine utilisée. Ce problème a bloqué la réalisation des tests visuels sur une partie du TP.
- **Compréhension du système de poids `layout_weight`** : La combinaison `layout_width="0dp"` avec `layout_weight` n'était pas intuitive au premier abord, notamment pour comprendre pourquoi une largeur nulle produit un meilleur résultat qu'une largeur `wrap_content`.
- **Ordre de déclaration dans le `RelativeLayout`** : Les références entre vues (`@id/vue`) imposent que la vue référencée soit déclarée avant celle qui l'utilise, ce qui a causé des erreurs lors des premiers essais.

### 9.2 Solutions appliquées

- **Migration vers un nouvel environnement** : Installation d'un dual boot sur un second ordinateur disposant de suffisamment de RAM. Le repos Git n'a pas encore été cloné sur la nouvelle machine afin de poursuivre le développement sans contrainte d'émulateur. Les TPs suivants seront réalisés sur cet environnement.
- **Apprentissage progressif des propriétés de layout** : Chaque propriété a été testée individuellement en observant son effet visuel dans l'éditeur graphique d'Android Studio, ce qui a permis de mieux comprendre les interactions entre `layout_weight`, `layout_width` et `layout_height`.
- **Nommage systématique des vues** : L'utilisation d'identifiants explicites (`android:id`) dès la création de chaque vue a permis d'éviter les erreurs de référencement dans les layouts relatifs.

## 10 Conclusion

Ce TP a permis d'explorer et de maîtriser les différents systèmes de mise en page proposés par Android, chacun répondant à des besoins spécifiques de conception d'interfaces graphiques.

Le **LinearLayout** s'est révélé simple et efficace pour organiser des composants en ligne ou en colonne. Le **TableLayout** offre une organisation structurée en lignes et colonnes, adaptée aux interfaces de type formulaire. Le **RelativeLayout** introduit un positionnement relatif entre les vues, offrant plus de flexibilité mais exigeant une attention particulière à l'ordre de déclaration des composants. Enfin, le **ConstraintLayout**, extension moderne du **RelativeLayout**, permet de concevoir des interfaces complexes et responsives grâce à un système de contraintes homogène et expressif.

Les concepts de `layout_weight`, `wrap_content`, `match_parent` et le système de contraintes relatif sont maintenant bien compris et peuvent être appliqués dans des projets plus complexes.