

## Solución Desafío - Elementos de la interfaz, navegación e interacción

### Requerimientos

1. Una actividad que contenga un NavHostFragment para mostrar los fragmentos.  
(1 Punto)

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior">

    <fragment
        android:id="@+id/nav_host_fragment_content_main"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:defaultNavHost="true"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/nav_graph" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

2. Pantalla principal definida en un fragmento donde el usuario ingresa la contraseña. El diseño de la pantalla es libre, pero debe incluir un campo de texto para ingresar la contraseña, un botón deshabilitado que se habilita cuando se cumplen los criterios y un texto indicando las características que debe tener la contraseña.  
(1 Punto)

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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"
```

```
tools:context=".FirstFragment">

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/passwordInputLayout"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginEnd="32dp"
    android:hint="@string/write_password"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.20">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/passwordEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword" />
</com.google.android.material.textfield.TextInputLayout>

<Button
    android:id="@+id/showButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="76dp"
    android:enabled="false"
    android:text="@string/validate_password"
    app:layout_constraintEnd_toEndOf="@+id/passwordInputLayout"
    app:layout_constraintStart_toStartOf="@+id/passwordInputLayout"
    app:layout_constraintTop_toBottomOf="@+id/passwordInputLayout" />

<TextView
    android:id="@+id/hiddenText"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="32dp"
    android:layout_marginEnd="32dp"
    android:text="@string/min_characters"
    android:visibility="visible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/showButton" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

3. Pantalla de resultado definida en un fragmento donde se le muestra al usuario los criterios usados para validar su contraseña y algunos consejos para mejorar su seguridad.
- (1 Punto)**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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"
    tools:context=".SecondFragment">

    <TextView
        android:id="@+id/textview_second"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="32dp"
        android:layout_marginEnd="32dp"
        android:text="@string/secure_password_description"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/secure_password_length_validation"
        app:layout_constraintStart_toStartOf="@+id/textview_second"
        app:layout_constraintTop_toBottomOf="@+id/textview_second" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/secure_password_uppercase_validation"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_marginTop="48dp"
        android:text="Otras características que hacen la contraseña más segura"
        app:layout_constraintStart_toStartOf="@+id/textView2"
        app:layout_constraintTop_toBottomOf="@+id/textView2" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/secure_password_character_validation"
    app:layout_constraintStart_toStartOf="@+id/textView3"
    app:layout_constraintTop_toBottomOf="@+id/textView3" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/secure_password_number_validation"
    app:layout_constraintStart_toStartOf="@+id/textView4"
    app:layout_constraintTop_toBottomOf="@+id/textView4" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

4. Utilizar ViewBinding para enlazar y utilizar las vistas. Debe estar definida en el build.gradle del módulo  
(1 Punto)

```
android {
    ...
    buildFeatures {
        viewBinding true
    }
}
```

y ser utilizada definida y utilizada en los fragmentos, por ejemplo, en el fragmento principal.

```
class FirstFragment : Fragment() {

    private var _binding: FragmentFirstBinding? = null

    private val binding get() = _binding!!

    override fun onCreateView(
```

```
inflater: LayoutInflater, container: ViewGroup?,
savedInstanceState: Bundle?
): View? {

    this._binding = FragmentFirstBinding.inflate(inflater, container, false)

    return binding.root
}
}
```

5. Validación de la contraseña aplicando los 2 criterios:

- a. Tiene más de 5 caracteres.
- b. Tiene al menos una letra mayúscula.

(1 Punto)

```
class FirstFragment : Fragment() {
    ...
    private fun validatePass(pass: String) {
        binding.showButton.isEnabled = validateLength(pass) && hasUpper(pass)
    }

    private fun validateLength(pass: String): Boolean = pass.length >= 5

    private fun hasUpper(pass: String): Boolean = pass.lowercase() != pass
}
```

6. Utilizar un ChangeListener para escuchar los cambios en la contraseña y activar el botón cuando se cumplen las condiciones. Cuando las condiciones no se cumplen, el botón debe estar deshabilitado.

(1 Punto)

```
class FirstFragment : Fragment() {

    private var _binding: FragmentFirstBinding? = null

    private val binding get() = _binding!!

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {

        this._binding = FragmentFirstBinding.inflate(inflater, container,
```

```
false)

    return binding.root

}

override fun onCreateView(view: View, savedInstanceState: Bundle?) {
    super.onCreateView(view, savedInstanceState)

    registerListener()
}

override fun onDestroyView() {
    super.onDestroyView()
    _binding = null
}

private fun registerListener() {
    binding.passwordEditText.addTextChangedListener(object :
    TextWatcher {
        override fun beforeTextChanged(cs: CharSequence, i: Int, i1:
    Int, i2: Int) {}
        override fun onTextChanged(charSequence: CharSequence, i:
    Int, i1: Int, i2: Int) {
            validatePass(charSequence.toString())
        }

        override fun afterTextChanged(editable: Editable) {}
    })
}
```

7. Utilizar `OnClickListener` para reaccionar al click en el botón.  
(1 Punto)

```
private fun registerListener() {

    binding.showButton.setOnClickListener {

    }
    . . .
}
```

8. Utilizar Navigation para navegar entre fragmentos. Debe estar definido el grafo de navegación y los fragmentos a utilizar.

(1 Punto)

```
<?xml version="1.0" encoding="utf-8"?>
<navigation 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:id="@+id/nav_graph"
    app:startDestination="@id/FirstFragment">

    <fragment
        android:id="@+id/FirstFragment"
        android:name="cl.dal.basicactivityexample.FirstFragment"
        android:label="@string/first_fragment_label"
        tools:layout="@layout/fragment_first">

        <action
            android:id="@+id/action_FirstFragment_to_SecondFragment"
            app:destination="@id/SecondFragment" />
    </fragment>
    <fragment
        android:id="@+id/SecondFragment"
        android:name="cl.dal.basicactivityexample.SecondFragment"
        android:label="@string/second_fragment_label"
        tools:layout="@layout/fragment_second">

        <action
            android:id="@+id/action_SecondFragment_to_FirstFragment"
            app:destination="@id/FirstFragment" />
    </fragment>
</navigation>
```

9. Navegar hacia la pantalla de resultado.

(1 Punto)

```
private fun registerListener() {

    binding.showButton.setOnClickListener {
        findNavController().navigate(R.id.action_FirstFragment_to_SecondFragment)
    }
    . . .
}
```

10. Extraer los textos al archivo de recursos de textos (strings.xml)  
(1 Punto)

```
<resources>
  <string name="app_name">BasicActivityExample</string>
  <string name="action_settings">Settings</string>
  <!-- Strings used for fragments for navigation -->
  <string name="first_fragment_label">Validador de contraseñas</string>
  <string name="second_fragment_label">Resultado</string>

  <string name="validate_password">Validar contraseña</string>
  <string name="min_characters">La contraseña debe tener al menos 5 caracteres
y una letra mayúscula</string>
  <string name="write_password">Escribe una contraseña</string>
  <string name="secure_password_description">Recuerda que para que la
contraseña sea segura debe cumplir con estas características:</string>
  <string name="secure_password_length_validation"> > Al menos 5
caracteres</string>
  <string name="secure_password_uppercase_validation"> > Al menos 1 letra
mayúscula</string>
  <string name="secure_password_number_validation"> > Al menos 1
número</string>
  <string name="secure_password_character_validation"> > Al menos 1 caracter
especial (como . , % $)</string>
</resources>
```