

1. Configura el Layout de la aplicación (aparece en la imagen que hay al final de este apartado) con:

a. Sólo debe haber un Layout y tiene que ser de tipo Constraint Layout.

b. Un ImageView

i. Debe de aparecer centrado horizontalmente.

ii. Debe mostrar la imagen que se proporciona “escudo.png”.

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:scaleType="centerCrop"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@mipmap/ic_escudo_foreground"
    tools:layout_editor_absoluteX="0dp" />
```

c. Un TextView con el texto “Selecciona Lugar”.

```
<TextView
    android:id="@+id/text_lugar"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="Selecciona Lugar"
    app:layout_constraintBottom_toBottomOf="@id/imageView"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView" />
```

d. Un Spinner

i. Cámbiale su id por otro que tú elijas.

ii. Alinear a la izquierda y debajo del ImageView anterior.

```
<Spinner
    android:id="@+id/sp_lugar"
    android:layout_width="170dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    app:layout_constraintLeft_toLeftOf="@id/imageView"
    app:layout_constraintTop_toBottomOf="@+id/imageView"
    tools:ignore="MissingConstraints" />
```

e. Un botón

i. Cámbiale su id por otro que tú elijas.

ii. El texto del botón será “Ver Localización”.

iii. El botón está alineado con el spinner anterior.

```

<Button
    android:id="@+id/btn_localizacion"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="44dp"
    android:text="VER LOCALIZACION"
    android:textColor="#000000"
    app:backgroundTint="#AEAEAE"
    app:layout_constraintLeft_toRightOf="@id/sp_lugar"
    app:layout_constraintTop_toBottomOf="@+id/imageView"
    tools:ignore="MissingConstraints" />

```

f. Un TextView con el texto "Precio Entrada", debajo del spinner.

```

<TextView
    android:id="@+id/text_precio"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    app:layout_constraintLeft_toLeftOf="@id/sp_lugar"
    app:layout_constraintTop_toBottomOf="@+id/sp_lugar"
    android:text="Precio Entrada" />

```

g. Un EditText, que aparezcan debajo del TextView anterior.

- i. Cambia el id por el que tú elijas.
- ii. El EditText debe estar deshabilitado para que no se pueda escribir en él ya que se va a rellenar automáticamente (propiedad enable).

```

<EditText
    android:id="@+id/edit_precio"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:ems="10"
    android:enabled="false"
    app:layout_constraintLeft_toLeftOf="@id/text_precio"
    app:layout_constraintTop_toBottomOf="@+id/text_precio" />

```

h. Un TextView con el texto "Nº de Entradas a reservar".

```

<TextView
    android:id="@+id/text_entradas"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    app:layout_constraintLeft_toLeftOf="@id/edit_precio"
    app:layout_constraintTop_toBottomOf="@+id/edit_precio"
    android:text="Nº de Entradas a reservar" />

```

- i. EditText, que aparezcan debajo del EditText anterior.
 - i. Cambia el id del EditText por el que tú elijas.
 - ii. El tipo de entrada del EditText debe ser un número.

```
<EditText
    android:id="@+id/edit_entradas"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="number"
    android:layout_marginTop="10dp"
    app:layout_constraintLeft_toLeftOf="@id/text_entradas"
    app:layout_constraintTop_toBottomOf="@+id/text_entradas" />
```

- j. Un TextView con el texto "Turno".

```
<TextView
    android:id="@+id/text_turno"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    app:layout_constraintLeft_toLeftOf="@id/edit_entradas"
    app:layout_constraintTop_toBottomOf="@+id/edit_entradas"
    android:text="Turno" />
```

- k. Dos RadioButtons

- i. Cámbiales su id por otros que tú elijas.
- ii. Deben tener el texto "Mañana" y "Tarde". Por defecto estará marcada la primera opción.
- iii. Aparecerá debajo del TextView anterior.

```

<RadioGroup
    android:id="@+id/radio_group"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    app:layout_constraintLeft_toLeftOf="@id/text_turno"
    app:layout_constraintTop_toBottomOf="@+id/text_turno">

    <RadioButton
        android:id="@+id/radio_mañana"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:text="Mañana"
        tools:layout_editor_absoluteX="97dp"
        tools:layout_editor_absoluteY="534dp" />

    <RadioButton
        android:id="@+id/radio_tarde"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tarde"
        tools:layout_editor_absoluteX="97dp"
        tools:layout_editor_absoluteY="534dp" />
</RadioGroup>

```

l. Un CheckBox

- i. Cambia el id por el que tú elijas.
- ii. Debe tener el texto "Recibir correo de confirmación".
- iii. Por defecto debe estar marcado.

```

<CheckBox
    android:id="@+id/check_correo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:checked="true"
    android:text="Recibir correo de confirmación"
    android:layout_marginTop="15dp"
    app:layout_constraintLeft_toLeftOf="@id/radio_group"
    app:layout_constraintTop_toBottomOf="@+id/radio_group" />

```

m. Un EditText.

- i. Cambia su id por el que tú elijas.

```
<EditText
    android:id="@+id/edit_correo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textEmailAddress"
    android:layout_marginTop="10dp"
    app:layout_constraintLeft_toLeftOf="@id/check_correo"
    app:layout_constraintTop_toBottomOf="@+id/check_correo" />
```

2. Funcionamiento del Spinner:

a. Deberá rellenar su contenido desde un array estático definido en un fichero de recursos.

```
<resources>
    <string name="app_name">Tarea21_Mendez_Alves_Alvaro</string>
    <string-array name="lugares">
        <item>Teatro del Mercado</item>
        <item>Casa de Cultura</item>
        <item>Piscina Municipal</item>
    </string-array>
</resources>
```

b. Las opciones que debe mostrar son: Teatro del Mercado, Casa de Cultura y Piscina Municipal.

c. No se podrá usar la propiedad entries para rellenar el Spinner. Se debe definir en el código un adaptador

```
ArrayAdapter<CharSequence> adaptadorLugares = ArrayAdapter.createFromResource(getApplication(),
    R.array.lugares, android.R.layout.simple_spinner_item);
lugares.setAdapter(adaptadorLugares);
adaptadorLugares.setDropDownViewResource(android.R.layout.simple_list_item_multiple_choice);
lugares.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        precio.setText(precios[position].toString());
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {
    }
});
```

Se crea el adaptador con los datos del array definido en el fichero de recursos. Se establece el adaptador del spinner lugares. Se capturan los eventos del spinner.

3. Funcionamiento del EditText Precio:

a. Siempre debe estar deshabilitado.

b. Se rellenará con el precio del lugar seleccionado en el Spinner.

c. Los precios de la entrada de cada lugar son:

i. Teatro del Mercado: 35

ii. Casa de Cultura: 8.

iii. Piscina Municipal: 2

```
Integer[] precios = new Integer[] {35, 8, 2};
```

Definición del array con los precios.

d. Los precios estarán definidos en un array estático en el código o en un fichero de recursos.

```
lugares.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
    @Override  
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
        precio.setText(precios[position].toString());  
    }  
  
    @Override  
    public void onNothingSelected(AdapterView<?> parent) {  
    }  
});
```

Cada vez que se seleccione un elemento del spinner, se actualizará el valor del precio en función a la posición del elemento seleccionado. Los precios tienen el mismo orden/posición que los lugares.

4. Funcionamiento del botón “Ver Localización”: Captura el evento `onClick` del botón (No se puede usar la propiedad `onClick`). Cuando se pulse en este botón, se abrirá el navegador con una búsqueda del lugar seleccionado. Se debe usar el intent implícito de búsqueda en Google. El texto de búsqueda debe ser, la opción seleccionada en el spinner a la que concatenaremos “Navalmoral de la Mata”. Si está seleccionado la opción Teatro del Mercado, el texto de búsqueda será: Teatro del Mercado Navalmoral de la Mata.

```
ver_localizacion.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        String lugar = lugares.getSelectedItem().toString() + " Navalmoral de la Mata";  
        Intent intent = new Intent(Intent.ACTION_WEB_SEARCH);  
        intent.putExtra(SearchManager.QUERY, lugar);  
        if (intent.resolveActivity(getPackageManager()) != null) {  
            startActivity(intent);  
        }  
    }  
});
```

Cuando se haga click sobre el botón `ver_localizacion`, se obtendrá el texto del ítem seleccionado en el spinner y se le concatenará la cadena “Navalmoral de la Mata”. Se creará un intent de búsqueda de Google.

5. Funcionamiento del EditText NumEntradas: Lo rellena el usuario de la aplicación con un número.

6. Funcionamiento de los Radiobuttons: Cada vez que se realice un cambio en el estado de alguno de los dos, se debe mostrar un Toast con el turno seleccionado.

```
radio_group.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {  
    public void onCheckedChanged(RadioGroup group, int checkedId) {  
        RadioButton checkedRadioButton = (RadioButton) group.findViewById(checkedId);  
  
        Toast toast = Toast.makeText(getApplicationContext(), checkedRadioButton.getText(),  
            duration: 100);  
        toast.show();  
    }  
});
```

Los dos radioButtons se encontrarán dentro de un radioGroup, cada vez que cambie la selección se mostrará un Toast con el texto del radioButton seleccionado.

7. Funcionamiento del EditText email:

a. El email aparecerá habilitado si el checkbox está marcado y deshabilitado si el checkbox está desmarcado.

```
check_correo.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
        if (buttonView.isChecked()) {  
            correo.setEnabled(true);  
        } else {  
            correo.setEnabled(false);  
        }  
    }  
});
```

Cuando cambia el estado del checkBox, se comprueba si está activado o no. Cuando esté activado, se habilita el EditText de correo, cuando está desactivado, este se deshabilita.

8. Crear un menú con dos opciones:

```
<?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/reservar"  
        android:title="RESERVAR"  
        app:showAsAction="ifRoom" />  
    <item android:id="@+id/cancelar"  
        android:title="CANCELAR"  
        app:showAsAction="ifRoom" />  
</menu>
```

Archivo menu.xml donde están definidas todas las entradas del menú.

```

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

```

Se añade el menú en el MainActivity.

a. Reservar:

i. Abrirá una nueva actividad con el detalle de la reserva (Ver la imagen al final de este apartado) La actividad principal debe pasar en el Intent los datos que se muestran en el detalle.

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    Toast toast;
    switch (item.getItemId()) {
        case R.id.reservar:
            Intent intent = new Intent( packageContext: MainActivity.this, ResumenActivity.class);
            intent.putExtra( name: "lugar", lugares.getSelectedItem().toString());
            intent.putExtra( name: "precio", precio.getText().toString());
            intent.putExtra( name: "entradas", entradas.getText().toString());
            intent.putExtra( name: "turno", ((RadioButton)findViewById(radio_group.
                getCheckedRadioButtonId())).getText().toString());
            MainActivity.this.startActivity(intent);
            return true;
        case R.id.cancelar:
            anularSelecciones();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

```

Cuando se seleccione la opción de reservar, se creará un intent en la que pasarán en el intent los valores "lugar", "precio", "entradas" y "turno" a la ResumenActivity.


```

public class ResumenActivity extends AppCompatActivity {

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

        Intent intent = getIntent();
        String lugar = intent.getStringExtra( name: "lugar");
        String precio = intent.getStringExtra( name: "precio");
        String entradas = intent.getStringExtra( name: "entradas");
        String turno = intent.getStringExtra( name: "turno");

        TextView text_lugar = findViewById(R.id.text_lugar);
        TextView text_precio = findViewById(R.id.text_precio);
        TextView text_entradas = findViewById(R.id.text_entradas);
        TextView text_turno = findViewById(R.id.text_turno);

        text_lugar.setText("Lugar: "+lugar);
        text_precio.setText("Precio Entrada: "+precio);
        text_entradas.setText("Nº Entradas Reservadas: "+entradas);
        text_turno.setText("Turno: "+turno);
    }
}

```

En ResumenActivity, existen cuatro TextViews, destinados a ser rellenos con la información recibida.

b. Cancelar: Anula las selecciones hechas por el usuario de la aplicación.

- i. La opción seleccionada del Spinner debe ser la primera (spinner.setSelection(posicion)).
- ii. Valor del EditText de número de entradas debe estar vacío.
- iii. Turno seleccionado, mañana.
- iv. CheckBox marcado
- v. Valor del EditText de número de entradas debe estar vacío.

```

private void anularSelecciones() {
    lugares.setSelection(0);
    entradas.setText("");
    radio_group.check(R.id.radio_mañana);
    check_correo.setChecked(true);
    correo.setText("");
}

```

Cuando se seleccione cancelar, se llamará a la función anularSelección, que restablecerá los valores por defecto.