



## Practica 4 – Agenda

### Objetivo

Realizar una agenda para guardar contactos y citas para determinada acción que se necesite, se utilizaran los controles:

- Spinner
- DatePickerDialog
- TimePickerDialog
- Intents
- ListView

### 1. Spinner

Permite al usuario seleccionar una única opción de un listado desplegable.

1. Abrir el archivo res->layouts->activity\_main.xml y agregar tres EditText, para nombre, apellido y teléfono (utilizar los tipos de EditText correspondiente)
2. Agrega un Spinner para seleccionar la categoría (Actividad Física, Trabajo, Compras, Recreativo, Otros)
3. Agregar dos EditText para la fecha y la hora
4. Agregar dos ImageButton para las acciones de la fecha y la hora (agregar recursos en mipmap)
5. El resultado debe quedar de la siguiente forma

El widget **Date Picker** se inicializa obteniendo los valores iniciales que usted configure, **DatePickerDialog** nos permite elegir y obtener la fecha con suma facilidad gracias a su interfaz gráfica.

Nombre

Apellido

Telefono

dd/mm/yyyy

hh:mm





## 6. Llenar el Spinner

### 6.1. Agregar el siguiente código en MainActivity.java

```
Spinner categoria;  
  
final String[] datos = new String[]{"Actividad Física", "Trabajo", "Compras",  
"Recreativo", "Otros"};  
  
ArrayAdapter<String> adaptador;
```

### 6.2. Dentro del método onCreate agregar el siguiente fragmento de código:

```
adaptador = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item,  
datos);  
  
categoria = (Spinner) findViewById(R.id.sp_category);  
  
categoria.setAdapter(adaptador);</layer-list>
```

## 7. Configurar el DatePickerDialog:

### 7.1. Hacer que la Activity implemente de **View.OnClickListener**

### 7.2. Agregar referencia a los controles EditText e ImageButton

```
EditText etFecha, etHora;  
  
ImageButton ibObtenerFecha, ibObtenerHora;
```

### 7.3. Referenciar los controles de layout con:

```
etFecha = (EditText) findViewById(R.id.et_mostrar_fecha_picker);  
etHora = (EditText) findViewById(R.id.et_mostrar_hora_picker);  
  
ibObtenerFecha = (ImageButton) findViewById(R.id.ib_obtener_fecha);  
ibObtenerHora = (ImageButton) findViewById(R.id.ib_obtener_hora);
```

### 7.4. Agregar el evento **setOnClickListener** a los controles ImageButton

### 7.5. Definir lógica del evento **OnClickListener**

## 8. Definir la lógica para obtener la fecha y la hora

### 8.1. Definir variables

```
private static final String CERO = "0";  
private static final String DOS_PUNTOS = ":";  
private static final String BARRA = "/";  
public final Calendar c = Calendar.getInstance();  
  
final int mes = c.get(Calendar.MONTH);  
final int dia = c.get(Calendar.DAY_OF_MONTH);  
final int anio = c.get(Calendar.YEAR);  
  
final int hora = c.get(Calendar.HOUR_OF_DAY);  
final int minuto = c.get(Calendar.MINUTE);
```

### 8.2. Codificar Lógica para extraer la fecha y la hora





## 2. Intent

### 1. Cambiar de Activity

- 1.1. Hacer que la Activity implemente de View.OnClickListener
- 1.2. Definir variables de controles

```
EditText txtName, txtLast, txtPhone;  
Button btnGuardar;
```

- 1.3. Agregar el findViewById a los controles previos

```
txtName = findViewById(R.id.txtName);  
txtLast = findViewById(R.id.txtLastName);  
txtPhone = findViewById(R.id.txtPhone);
```

- 1.4. Crear Actividad **ListDatesActivity.java** para redireccionar el **MainActivity**

- 1.5. Agregar un método que hará el cambio de Activity

```
Intent i = new Intent(this, ListDatesActivity.class);  
startActivity(i);
```

### 2. Pasar parámetros a un activity

3. Crear una clase Citas para capturar los datos de **MainActivity.java** la clase de implementar de **Serializable**

4. Agregar al método donde se agregó el intent **i.putExtra("Cita", getDataUI());**

5. En ListDateActivity.java agregar el siguiente código en método onCreate

```
Cita c = (Cita) getIntent().getSerializableExtra("Cita");  
Toast.makeText(this, c.getCategory(), Toast.LENGTH_LONG).show();
```



