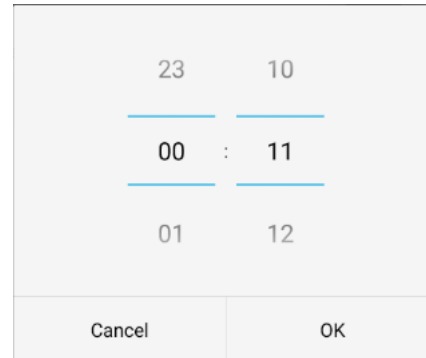
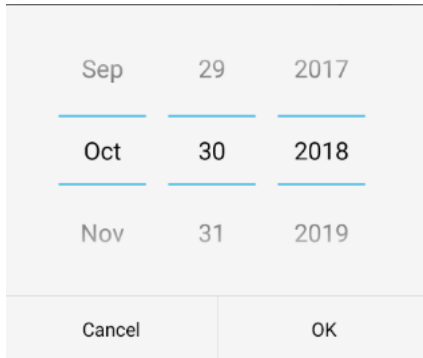


Selectores de fecha y/u hora

En inglés se llaman *Pickers*, *DatePicker* o *TimePicker*. El uso de estos selectores te evitará muchas comprobaciones en código sobre si el usuario escribió correctamente el formato de hora/fecha o si se ajustan a las configuraciones locales del dispositivo móvil o si es una fecha válida



Al igual que los Diálogos, y aunque se pueden usar directamente en un Layout, Android también recomienda utilizar la clase `DialogFragment` para implementarlos (ya visto en clase)

El caso práctico consiste en un sencillo formulario que muestra al usuario la posibilidad de introducir su fecha de nacimiento y una hora a través de un campo de texto (*EditText*). Estos campos de fecha y hora estarán inhabilitados para forzar al usuario a pulsar un botón para poder seleccionar de un *Picker* tanto la fecha como la hora. Al pulsar en cada respectivo botón se le mostrará un diálogo con su *Picker* correspondiente:

Para cada diálogo creamos una clase que herede de `DialogFragment` y que poseerá una interfaz para comunicarse con la actividad principal, devolviendo esta un objeto `Fecha` cuando se haya seleccionado tanto la fecha como la hora. Para la devolución utilizaremos la clase `GregorianCalendar`, que sustituye a la clase `Date` de Java al quedar esta depreciada (*deprecated*) en la API 1. Para la creación del diálogo se puede usar un Layout XML que contenga un *Picker* o directamente utilizar las clases `DatePickerDialog` y `TimePickerDialog` que directamente construyen un *Dialog* con un *Picker*. En nuestro ejemplo usaremos esta segunda opción. Además, al igual que en el ejemplo de `DialogFragment`, al crear el fragmento se ejecutará el método `onAttach`, momento que aprovecharemos para quedarnos con una referencia a la actividad y así poder invocar al método `onResultadoFecha/onResultadoHora` de la interfaz y pasar el resultado de la elección del usuario a la actividad principal.



La clase `DialogoFecha` consistirá en:

```
public class DialogoFecha extends DialogFragment implements
DatePickerDialog.OnDateSetListener{

    OnFechaSeleccionada f;
    @Override
    public void onAttach(Activity activity) {
        f=(OnFechaSeleccionada) activity;
        super.onAttach(activity);
    }
}
```

```

@Override
public Dialog onCreateDialog(Bundle savedInstanceState) {
    Calendar c=Calendar.getInstance();
    int año=c.get(Calendar.YEAR);
    int mes=c.get(Calendar.MONTH);
    int dia=c.get(Calendar.DAY_OF_MONTH);

    return new DatePickerDialog(getActivity(), this, año, mes, dia);
}

@Override
public void onDateSet(DatePicker datePicker, int i, int i2, int i3) {
    GregorianCalendar g=new GregorianCalendar(i,i2,i3);
    f.onResultadoFecha(g);
}

public interface OnFechaSeleccionada{
    public void onResultadoFecha(GregorianCalendar fecha);
}
}

```

Y la clase DialogoHora es muy similar:

```

public class DialogoHora extends DialogFragment implements
TimePickerDialog.OnTimeSetListener{

    OnHoraSeleccionada f;
    @Override
    public void onAttach(Activity activity) {
        f=(OnHoraSeleccionada) activity;
        super.onAttach(activity);
    }

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        Calendar c=Calendar.getInstance();
        int hora=c.get(Calendar.HOUR);
        int minutos=c.get(Calendar.MINUTE);

        return new TimePickerDialog(getActivity(), this, hora, minutos, true);
    }

    @Override
    public void onTimeSet(TimePicker timePicker, int i, int i2) {
        GregorianCalendar g=new GregorianCalendar();
        g.set(Calendar.HOUR, i);
        g.set(Calendar.MINUTE, i2);
        f.onResultadoHora(g);
    }

    public interface OnHoraSeleccionada{
        public void onResultadoHora(GregorianCalendar hora);
    }
}

```

Al crear un objeto de estos tipos DialogoFecha/DialogoHora e invocar a su método show(), se ejecutará el método *onCreateDialog()* que debe retornar el diálogo. Para crear el diálogo usamos los constructores:

```

new DatePickerDialog(getActivity(), this, año, mes, dia);
new TimePickerDialog(getActivity(), this, hora, minutos, true);

```

A estos constructores, les pasamos los valores de fecha y hora actuales (año,mes,día) y (hora,minutos) obtenidos la clase Calendar. La clase Calendar nos proporciona acceso a la fecha y hora actual del dispositivo móvil. Para formar una fecha y una hora a partir de la elección del usuario usamos los métodos de *callback on<X>Set* de las interfaces *OnXSetListener*, donde <X> puede ser *Time* o *Date*. Estos métodos nos avisan de que el usuario ha definido una nueva fecha y hora en el diálogo y ha retornado con éxito de la selección. En ese momento, se construye un objeto *GregorianCalendar* con los valores seleccionados por el usuario (estos se reciben en los parámetros de la función *On<X>Set*) y se invoca a la función *onResultadoHora/onResultadoFecha* de la interfaz correspondiente y que debe implementar la actividad. Estos dos últimos métodos reciben como parámetro un objeto *GregorianCalendar*, subclase de *Calendar*.

En los siguientes enlaces puedes ver más información sobre ambas clases:

<http://developer.android.com/reference/java/util/Calendar.html>

<http://developer.android.com/reference/java/util/GregorianCalendar.html>

Por último, la actividad principal, tendrá 4 funciones programadas:

```
public void onClickFecha(View view) {
    DialogoFecha d=new DialogoFecha();
    d.show(getFragmentManager(),"Mi diálogo Fecha");
}

public void onClickHora(View view){
    DialogoHora d=new DialogoHora();
    d.show(getFragmentManager(),"Mi diálogo Hora");
}

@Override
public void onResultadoFecha(GregorianCalendar fecha) {
    EditText et=(EditText)findViewById(R.id.etFechaNacimiento);
    et.setText(fecha.get(Calendar.DAY_OF_MONTH)+"/"+(fecha.get(Calendar.MONTH)+1)
        +"/"+fecha.get(Calendar.YEAR));
}

@Override
public void onResultadoHora(GregorianCalendar hora) {
    EditText et=(EditText)findViewById(R.id.etHora);
    et.setText(hora.get(Calendar.HOUR)+":"+hora.get(Calendar.MINUTE));
}
```

Las funciones *onClickFecha* y *onClickHora* son los callback de los botones que pulsa el usuario cuando quiere cambiar la fecha, y *onResultadoFecha* y *onResultadoHora* son las funciones de callback que hay que implementar por la necesidad de comunicarse con el *DialogFragment* correspondiente.

Y por tanto recuerda:

```
public class MyActivity extends Activity implements
    DialogoFecha.OnFechaSeleccionada, DialogoHora.OnHoraSeleccionada{
```