www.centic.es

Curso Android
Edición **2011**

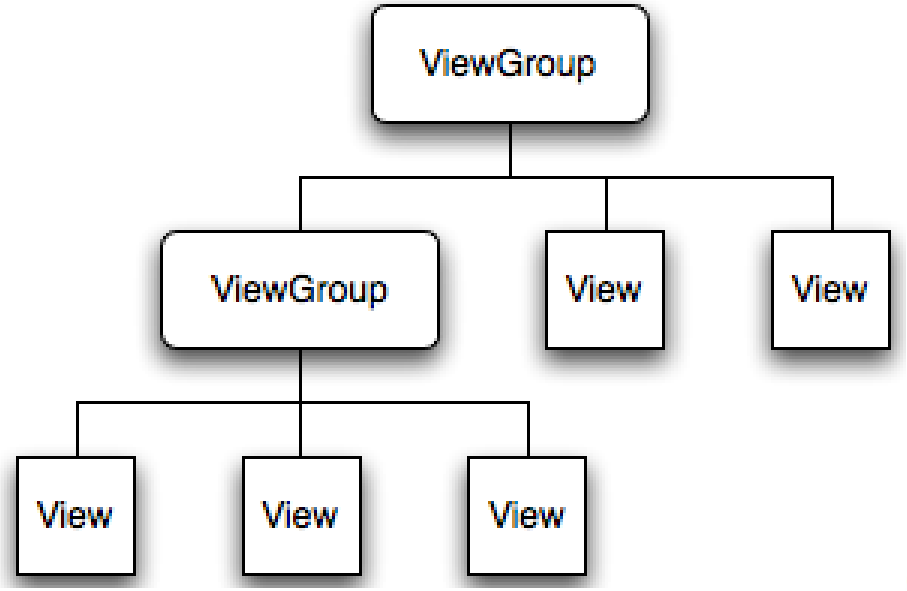
CENTRO TECNOLÓGICO DE LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

centic

Interfaz de Usuario.

Jerarquía de componentes.

- Árbol de componentes
  - View
  - ViewGroup



centic

Interfaz de Usuario.

ViewGroup.

- Layout
  - Son las ramas del árbol de la GUI

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent"
              android:orientation="vertical" >
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button" />
</LinearLayout>
```

Interfaz de Usuario.

View.

- Widgets
  - Son las hojas del árbol de la GUI

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent"
              android:orientation="vertical" >
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button" />
</LinearLayout>
```

centic

Interfaz de Usuario.

Eventos de la GUI.

- Registrar eventos
- setOnTouchListener()
- setOnClickListener()
- setOnKeyListener()

```
btn.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub

    }
});
```

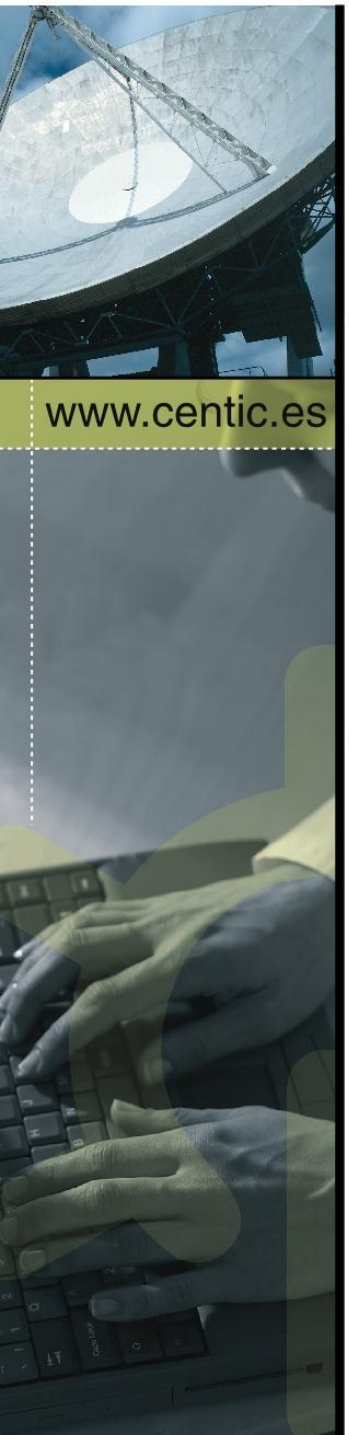
centic

Interfaz de Usuario.

Eventos de la GUI.

- Sobreescribir eventos
- OnTouchListener()
- OnClickListener()
- OnKeyListener()

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    // TODO Auto-generated method stub
    return super.onTouchEvent(event);
}
```



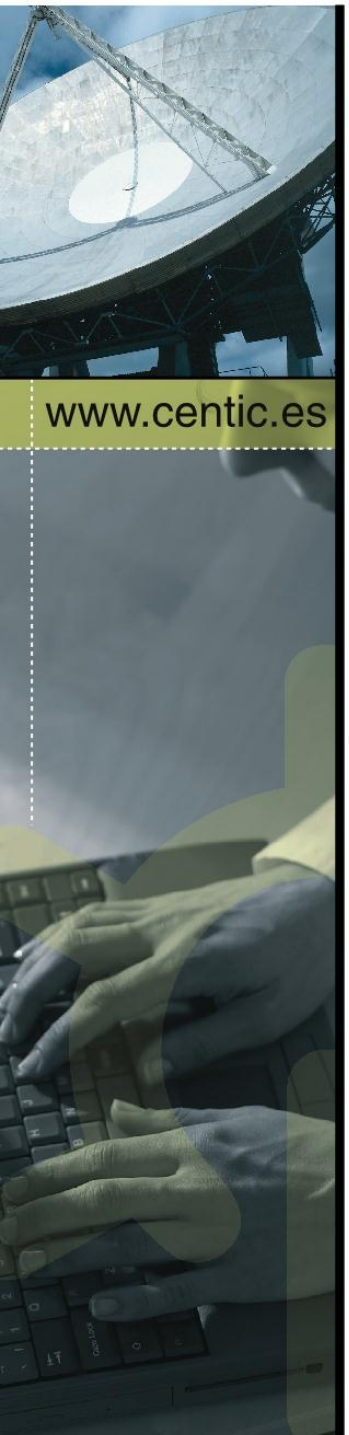
Interfaz de Usuario.

Menus.

- Menu (Key Menu)

- Context Menu

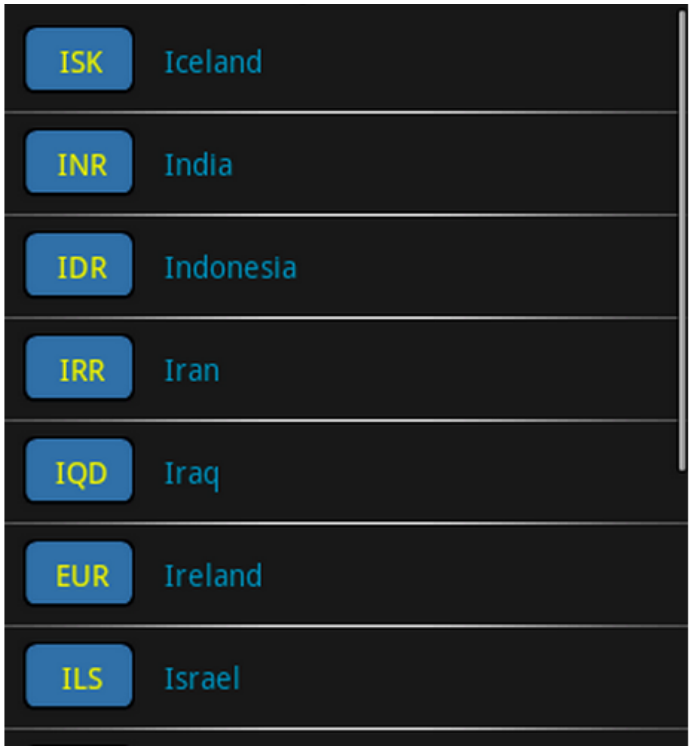
centic



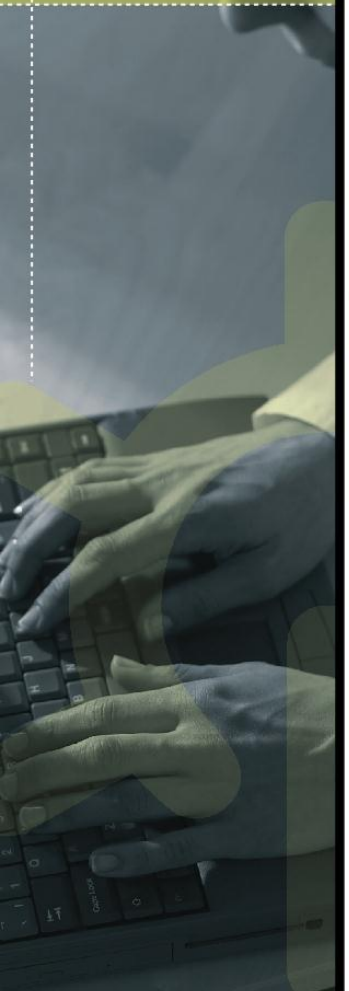
Interfaz de Usuario.

Adaptadores.

- AdapterView
  - Listview
  - Spinner



centic



Interfaz de Usuario.

Declarar Layouts.

- En código

```
//Creamos el layout
LinearLayout llPrincipal = new LinearLayout(this);
LayoutParams lpParams = new LayoutParams(LayoutParams.FILL_PARENT,
                                         LayoutParams.FILL_PARENT);
llPrincipal.setLayoutParams(lpParams);

//Añadimos los componentes que contiene
//el layout
llPrincipal.addView(new TextView(this));

//Añadimos el layout a la activity
setContentView(llPrincipal);
```

centic

Interfaz de Usuario.

Declarar Layouts.

- Desde XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent"
              android:orientation="vertical" >
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button" />
</LinearLayout>
```

Interfaz de Usuario.

Declarar Layouts.

- Desde XML

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main_layout);
}
```

Interfaz de Usuario.

Atributos.

- Android:id
- El id debe ser único a todos los recuros
- tipoNombreLayout «llVerContactosMiscontactos»
- @tipo indica un recurso existente
- @+id indica un nuevo recurso

```
android:id="@+id/my_button"
```

```
android:id="@android:id/empty"
```

Interfaz de Usuario.

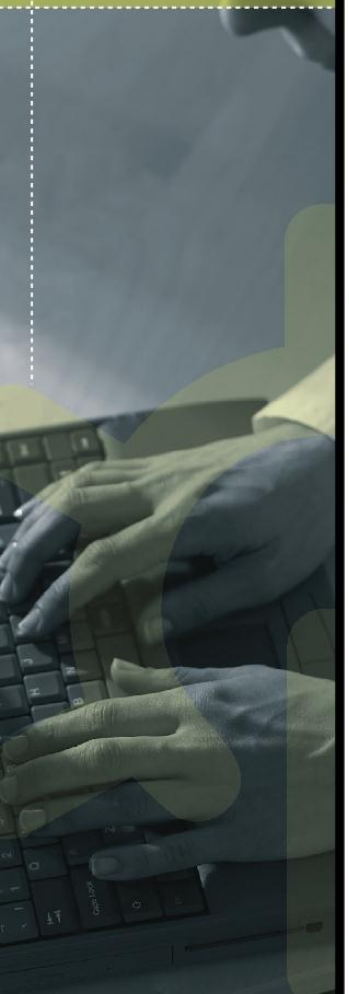
Atributos.

- Recuperar componentes

```
<Button android:id="@+id/my_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/my_button_text"/>
```

```
Button myButton = (Button) findViewById(R.id.my_button);
```

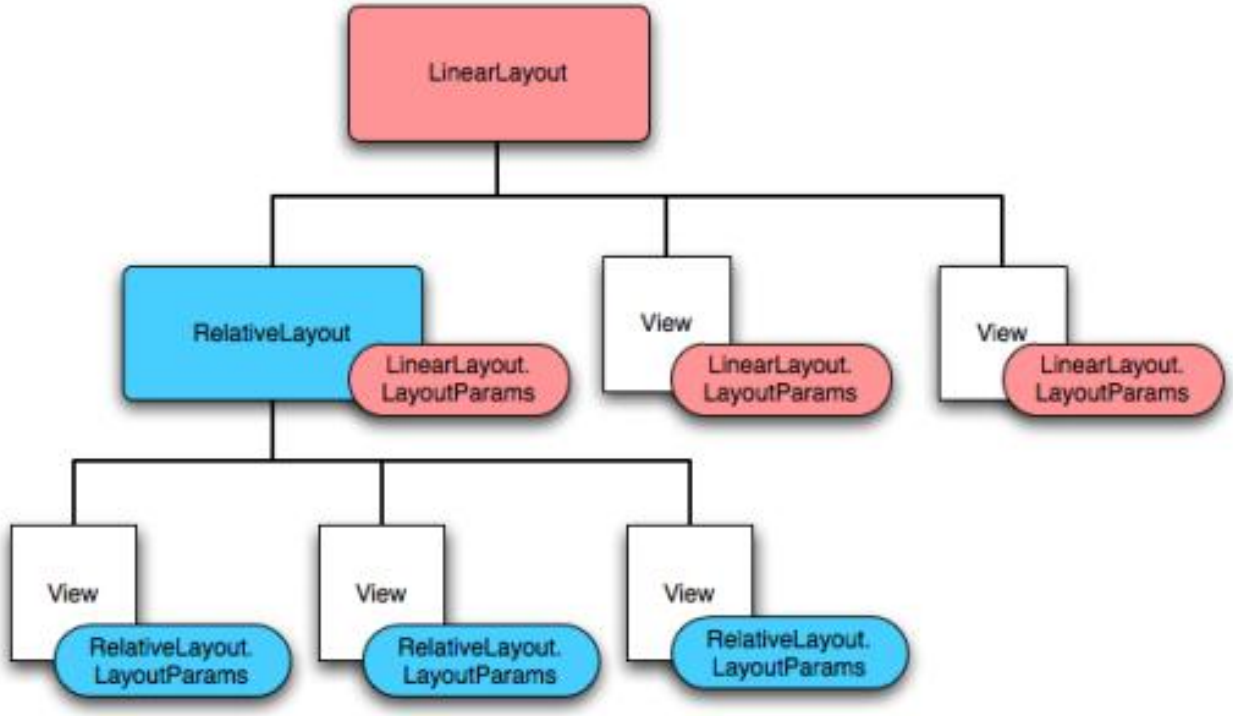
centic



Interfaz de Usuario.

Atributos.

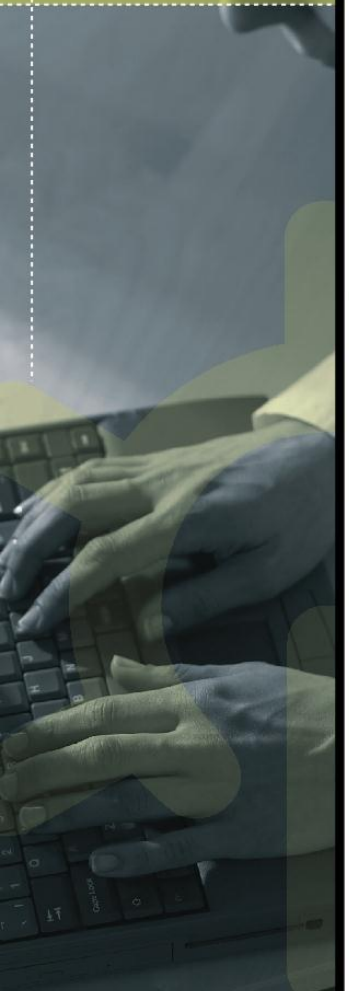
- Android:Layout_*



Interfaz de Usuario.

Atributos.

- Márgenes externos
  - Layout_margin
  - Layout_marginXXX
    - Layout_marginLeft
- Márgenes internos
  - Padding
  - paddinXXX
    - paddingLeft



Interfaz de Usuario.

Atributos.

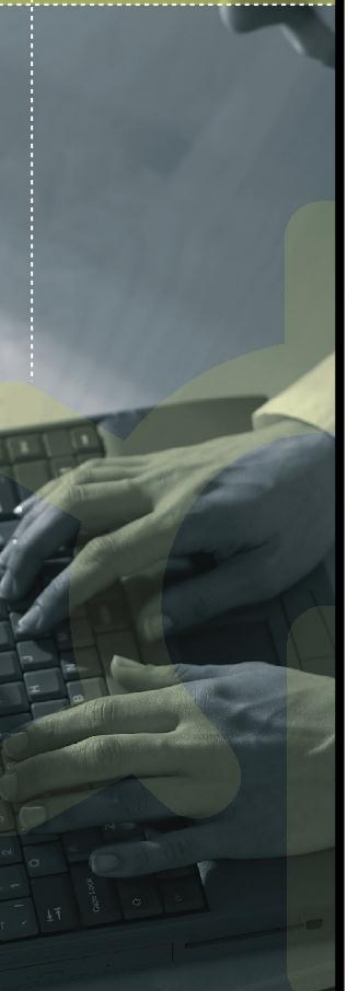
- LayoutParams (XML)
  - Layout_width
  - Layout_height
  - Wrap_content
  - Match_parent (fill_parent < 8 (2.2))
  - Valor concreto
- Unidades
  - Pixel «px» Evitar!!!!
  - Independent Pixel Units «dp»

Interfaz de Usuario.

Atributos.

- Posición
  - Left
  - Top
  - getLeft()
  - getTop()
  - getRight()
  - getBottom()
- Equivalencias
  - getRight() = getLeft() + getWidth()
  - getBottom = getTop() + getHeight()

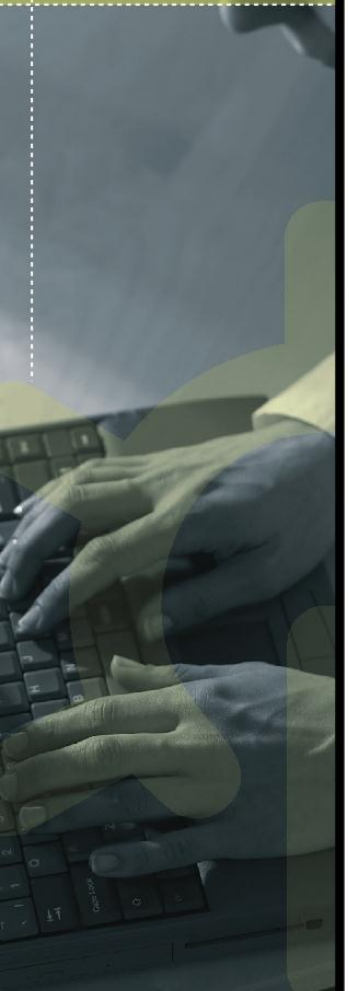
centic



Interfaz de Usuario.

Atributos.

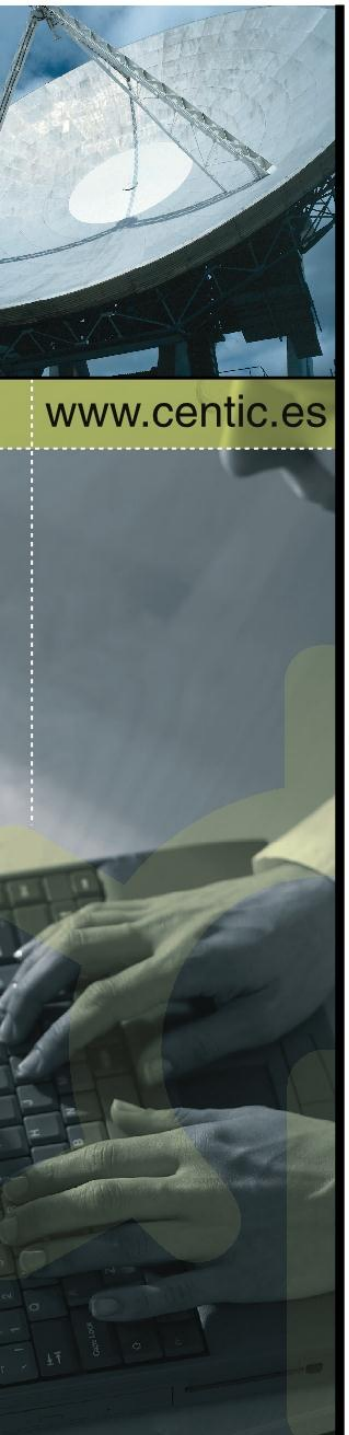
- Tamaño deseado.
  - Measured width
  - getMeasureWidth()
  - Measured height
  - getMeasureHeight()
- Tamaño actual
  - Width
  - getWidth()
  - Height
  - getHeight()



Interfaz de Usuario.

Atributos.

- Padding. Margen interior
  - setPadding(left, top, right, bottom)
  - setPaddingXXX()
  - getPaddingXXX()
- Margin. Margen exterior
  - setMargins(left, top, right, bottom)
  - getMarginXXX()
  - setMarginXXX()



Interfaz de Usuario.

Tipos de Layouts.

- FrameLayout
  - Es el más básico
  - Los elementos contenidos se posicionan «LEFT/TOP»
  - El contenido se superpone
  - Es el primer elemento de la vista «ROOT Element»

centic

Interfaz de Usuario.

Tipos de Layouts.

- LinearLayout
  - Alinea todo el contenido en una dirección
  - android:orientation
    - Horizontal
    - Vertical
  - Su contenido es posicionado uno a continuación de otro, siguiendo el orden en que fueron declarados
  - Permite usar márgenes internos y externos
    - android:ayout_marginXXX
    - android:paddinXXX

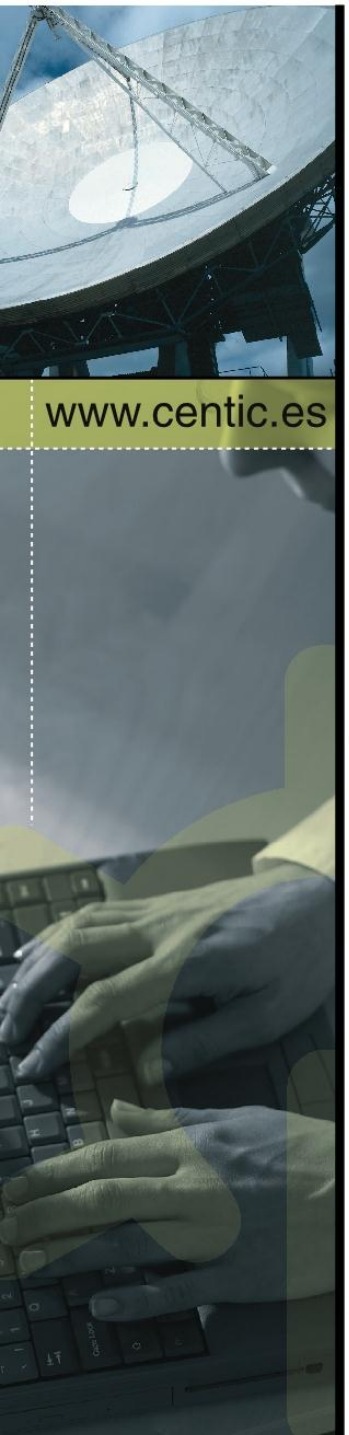
centic

Interfaz de Usuario.

Tipos de Layouts.

- LinearLayout
  - Permite usar «android:gravity»
    - Center_vertical
    - Center_horizontal
    - Center_vertical | center_horizontal
  - Permite especificar una importancia al contenido
    - Android:layout_weight (0 - X)

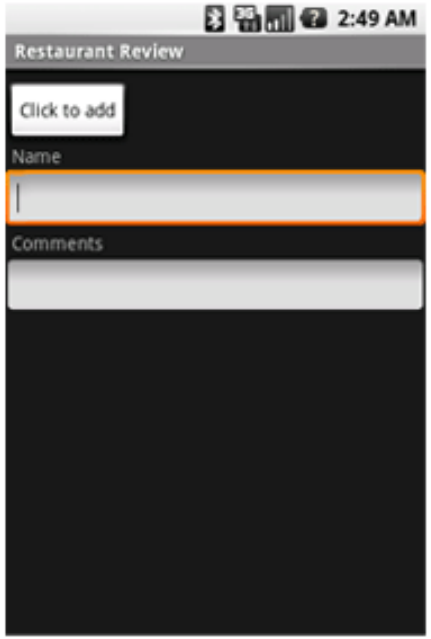
centic

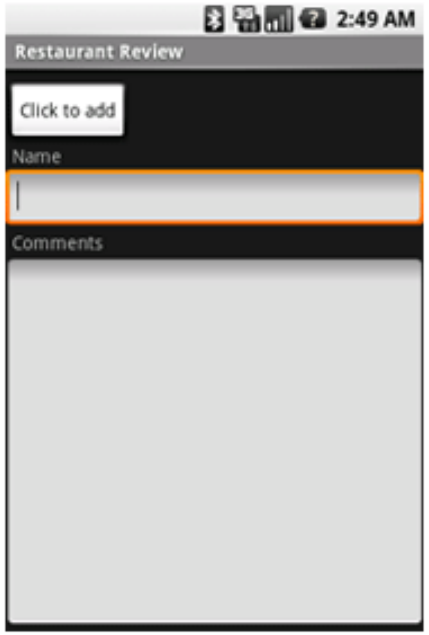


Interfaz de Usuario.

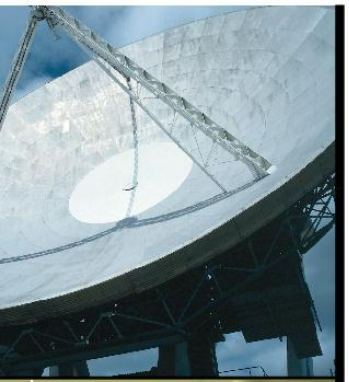
Tipos de Layouts.

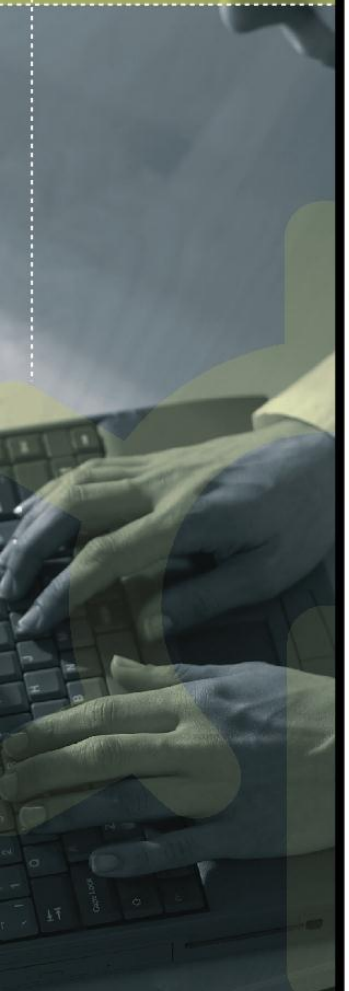
- LinearLayout





centic





Interfaz de Usuario.

Tipos de Layouts.

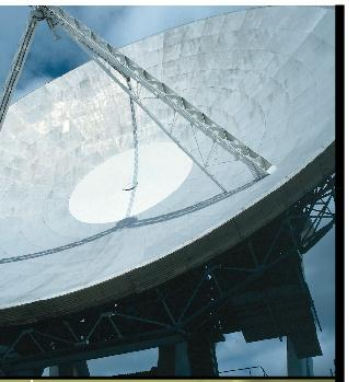
- TableLayout
  - Posiciona su contenido en filas y columnas
  - Podemos dejar celdas vacias
  - Podemos englobar celdas como en HTML
  - Cada fila se define en un TableRow
  - Cada TableRow puede tener 0 a n celdas
  - Cada celda almacena una VIEW
  - Podemos hacer las celdas retráctiles
    - setColumnShrikable()
  - Podemos hacer las celdas expandibles
    - setColumnStretchable()

Interfaz de Usuario.

Tipos de Layouts.

- TableLayout
  - Posiciona su contenido en filas y columnas
  - Podemos dejar celdas vacías
  - Podemos englobar celdas como en HTML
  - Cada fila se define en un TableRow
  - Cada TableRow puede tener 0 a n celdas
  - Cada celda almacena una VIEW

centic



Interfaz de Usuario.

Tipos de Layouts.

- TableLayout
  - Podemos hacer las celdas retráctiles
    - setColumnShrikable()
  - Podemos hacer las celdas expandibles
    - setColumnStretchable()
  - Android:layout_column
    - Indicamos el numero de coulmna
    - Incremento automático

Interfaz de Usuario.

Tipos de Layouts.

- TableLayout

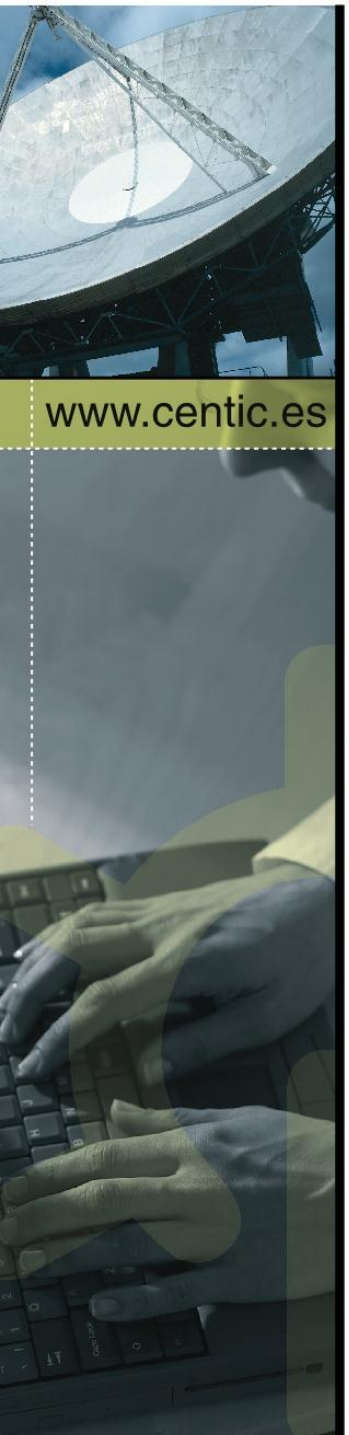
```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:stretchColumns="1">

    <TableRow>
        <TextView
            android:layout_column="1"
            android:text="Open..."
            android:padding="3dip" />
        <TextView
            android:text="Ctrl-O"
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>
```

Interfaz de Usuario.

Tipos de Layouts.

- TableLayout

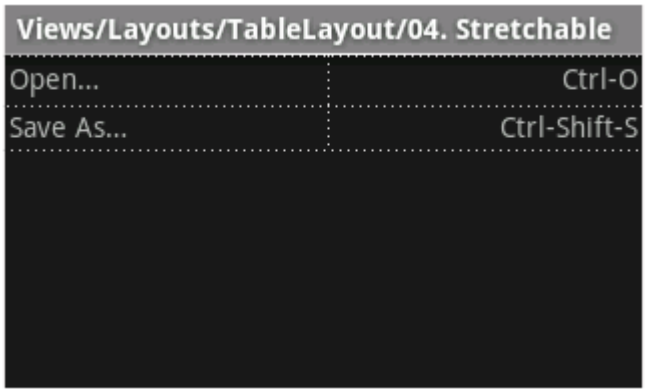
```
<TableRow>
    <TextView
        android:layout_column="1"
        android:text="Save..."
        android:padding="3dip" />
    <TextView
        android:text="Ctrl-S"
        android:gravity="right"
        android:padding="3dip" />
</TableRow>

<TableRow>
    <TextView
        android:layout_column="1"
        android:text="Save As..."
        android:padding="3dip" />
    <TextView
        android:text="Ctrl-Shift-S"
        android:gravity="right"
        android:padding="3dip" />
</TableRow>
```



Interfaz de Usuario.

Tipos de Layouts.

- TableLayout



centic

Interfaz de Usuario.

Tipos de Layouts.

- RelativeLayout
- Especificamos la posición de los componentes en relación a su contenedor o a otros componentes.

```
android:layout_toLeftOf="@id/otrocomponente"

android:layout_toRightOf="@id/otrocomponente"

android:layout_above="@id/otrocomponente"

android:layout_below="@id/otrocomponente"
```

centic

Interfaz de Usuario.

Tipos de Layouts.

- RelativeLayout

```
android:layout_alignBaseline="@id/otrocomponente"

android:layout_alignLeft="@id/otrocomponente"

android:layout_alignRight="@id/otrocomponente"

android:layout_alignBottom="@id/otrocomponente"
```

centic

Interfaz de Usuario.

Tipos de Layouts.

- RelativeLayout

```
android:layout_alignParentLeft="True"

android:layout_alignParentTop="True"

android:layout_alignParentRight="True"

android:layout_alignParentBottom="True"

android:layout_centerInParent="True"
```

centic

Interfaz de Usuario.

Tipos de Layouts.

- RelativeLayout

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:background="@drawable/blue"
                android:padding="10px" >

    <TextView android:id="@+id/label"
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:text="Type here:" />
```

centic

Interfaz de Usuario.

Tipos de Layouts.

- RelativeLayout

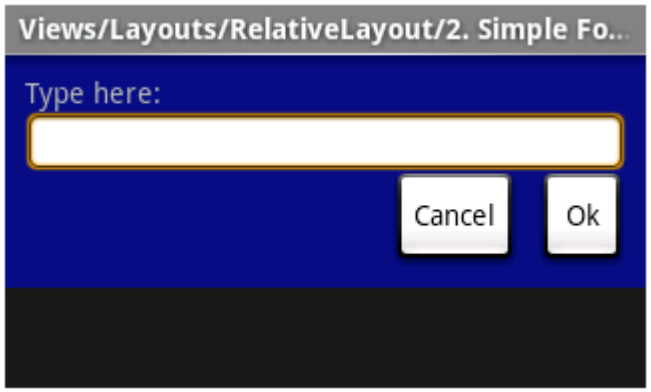
```
<EditText android:id="@+id/entry"
          android:layout_width="fill_parent"
          android:layout_height="wrap_content"
          android:background="@android:drawable/editbox_background"
          android:layout_below="@id/label" />

<Button android:id="@+id/ok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/entry"
        android:layout_alignParentRight="true"
        android:layout_marginLeft="10px"
        android:text="OK" />
```

Interfaz de Usuario.

Tipos de Layouts.

- RelativeLayout

```
    <Button android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_toLeftOf="@id/ok"
            android:layout_alignTop="@id/ok"
            android:text="Cancel" />
</RelativeLayout>
```



centic

Interfaz de Usuario.

Tipos de Layouts.

- FrameLayout
- Gallery
- GridView
- LinearLayout
- ListView
- RelativeLayout
- ScrollView
- Spinner
- SurfaceView
- TabHost
- TableLayout
- ViewFlipper
- ViewSwitcher

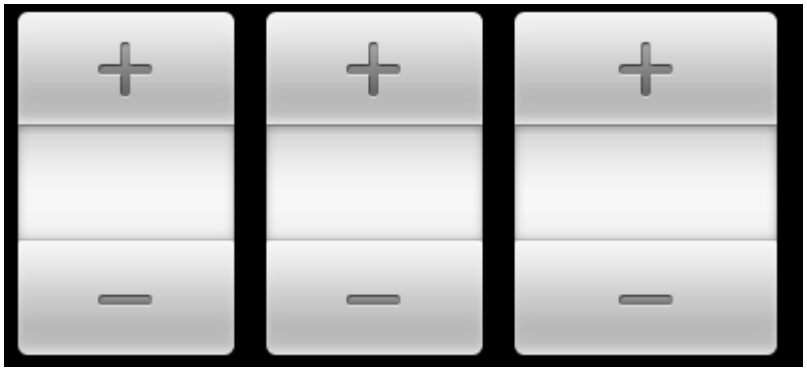
centic

Interfaz de Usuario.

Widgets.

- DatePicker

```
<DatePicker
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:startYear="2000"
    android:endYear="2020"
    android:id="@+id/dpFecha"
/>
```

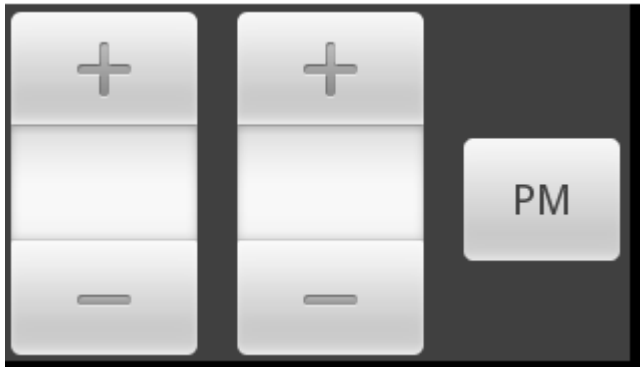


Interfaz de Usuario.

Widgets.

- TimePicker

```
<TimePicker
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/dpFecha"
/>
```



centic

Interfaz de Usuario.

Widgets.

- Button

```
<Button
    android:text="Ejemplo Boton!!!"
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

centic

Interfaz de Usuario.

Widgets.

- Button

```
final Button button = (Button) findViewById(R.id.button);
button.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        // Perform action on clicks
        Toast.makeText(HelloFormStuff.this, "Beep Bop", Toast.LENGTH_SHORT).show();
    }
});
```

Interfaz de Usuario.

Widgets.

- EditText

```
<EditText
    android:id="@+id/etTexto"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"/>
```

Interfaz de Usuario.

Widgets.

- EditText

```
et.setText("Texto a mostrar");
String texto = et.getText().toString();

et.setOnEditorActionListener(new OnEditorActionListener() {

    public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {
        // TODO Auto-generated method stub
        return false;
    }
});
```

centic

Interfaz de Usuario.

Widgets.

- TextView

```
<TextView
    android:id="@+id/tvMensaje"
    android:text="@string/hello"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

Mensaje a mostrar. Esto debe estar en @string

centic

Interfaz de Usuario.

Widgets.

- TextView

```
TextView tv = (TextView)findViewById(R.id.tvMensaje);
tv.setText(R.string.hello);
```

Mensaje a mostrar. Esto debe estar en @string

Interfaz de Usuario.

Widgets.

- Checkbox

```
<CheckBox android:id="@+id/checkbox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="check it out" />
```

Interfaz de Usuario.

Widgets.

- Checkbox

```
CheckBox ck = (CheckBox)findViewById(R.id.checkbox);
ck.setOnCheckedChangeListener(new OnCheckedChangeListener() {

 public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
     // TODO Auto-generated method stub

 }
});
```

Interfaz de Usuario.

Widgets.

- RadioButton

```
<RadioGroup
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:orientation="vertical">
  <RadioButton android:id="@+id/radio_red"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Red" />
  <RadioButton android:id="@+id/radio_blue"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Blue" />
</RadioGroup>
```

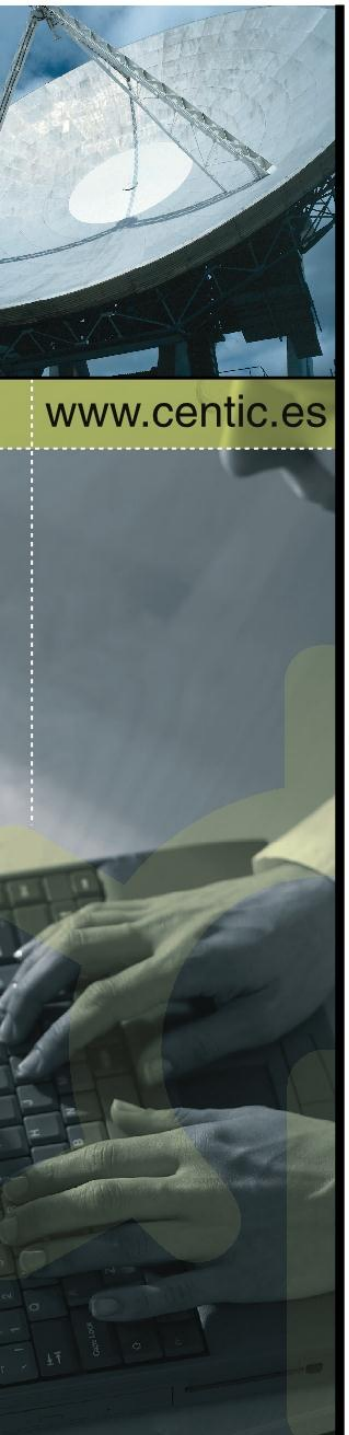
Interfaz de Usuario.

Widgets.

- RadioButton

```
private OnClickListener radio_listener = new OnClickListener() {
    public void onClick(View v) {
        // Perform action on clicks
        RadioButton rb = (RadioButton) v;
        Toast.makeText(HelloFormStuff.this, rb.getText(), Toast.LE
    }
};
```

```
final RadioButton radio_red = (RadioButton) findViewById(R.id.radio_red);
final RadioButton radio_blue = (RadioButton) findViewById(R.id.radio_blue);
radio_red.setOnClickListener(radio_listener);
radio_blue.setOnClickListener(radio_listener);
```

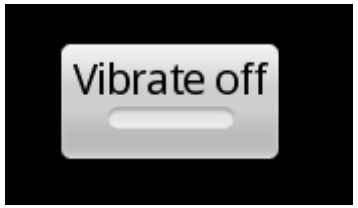


Interfaz de Usuario.

Widgets.

- ToggleButton

```
<ToggleButton android:id="@+id/togglebutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOn="Vibrate on"
    android:textOff="Vibrate off"/>
```



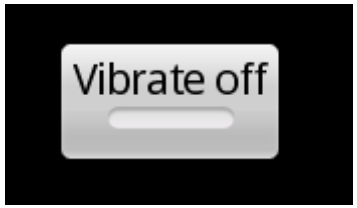
centic

Interfaz de Usuario.

Widgets.

- ToggleButton

```
final ToggleButton togglebutton = (ToggleButton) findViewById(R.id.togglebutton);
togglebutton.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        // Perform action on clicks
        if (togglebutton.isChecked()) {
            Toast.makeText(HelloFormStuff.this, "Checked", Toast.LENGTH_SHORT).sho
        } else {
            Toast.makeText(HelloFormStuff.this, "Not checked", Toast.LENGTH_SHORT
        }
    }
```



centic

Interfaz de Usuario.

Widgets.

- RatingBar

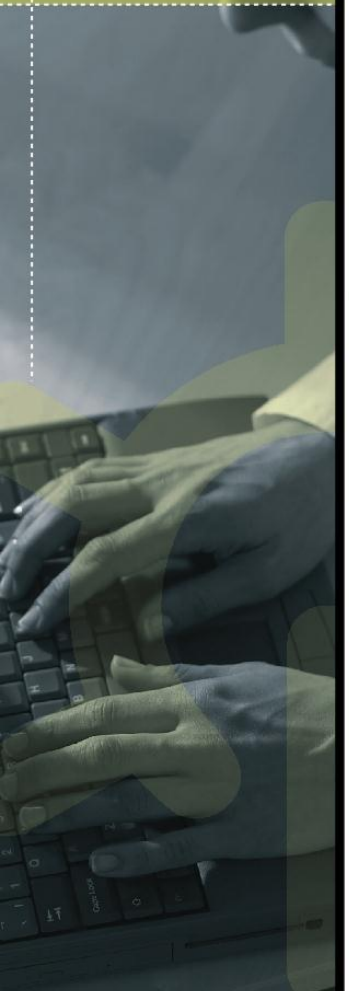
```
<RatingBar android:id="@+id/ratingbar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:numStars="5"
    android:stepSize="1.0"/>
```

Interfaz de Usuario.

Widgets.

- RatingBar

```
RatingBar ratingbar = (RatingBar) findViewById(R.id.ratingbar);
ratingbar.setOnRatingBarChangeListener(new OnRatingBarChangeListener() {
    public void onRatingChanged(RatingBar ratingBar, float rating, boolean fromUser) {


    }
});
```



Interfaz de Usuario.

Eventos.

- Listeners
  - onClick()
  - onLongClick()
  - onFocusChanged()
  - onKey()
  - onTouch()
  - onCreateContextMenu()

Interfaz de Usuario.

Eventos.

- Listeners

```
// Create an anonymous implementation of OnClickListener
private OnClickListener mCorkyListener = new OnClickListener() {
    public void onClick(View v) {
      // do something when the button is clicked
    }
};

protected void onCreate(Bundle savedValues) {
    ...
    // Capture our button from layout
    Button button = (Button)findViewById(R.id.corky);
    // Register the onClick listener with the implementation above
    button.setOnClickListener(mCorkyListener);
    ...
}
```

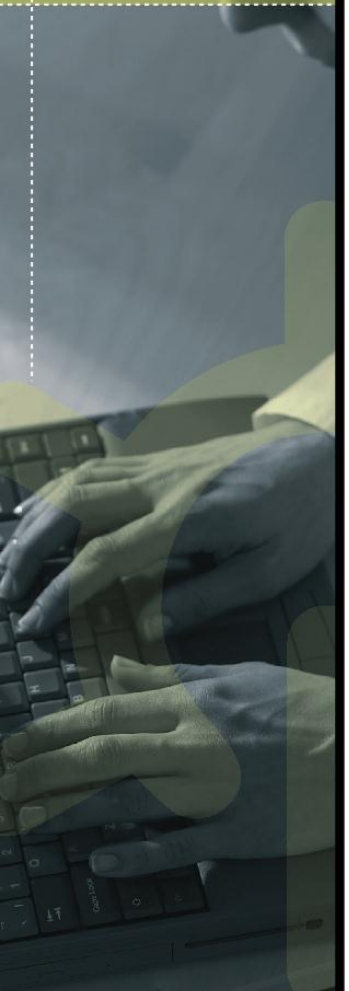
Interfaz de Usuario.

Eventos.

- Listeners

```
public class ExampleActivity extends Activity implements OnClickListener {
    protected void onCreate(Bundle savedValues) {
        ...
        Button button = (Button)findViewById(R.id.corky);
        button.setOnClickListener(this);
    }

    // Implement the OnClickListener callback
    public void onClick(View v) {
      // do something when the button is clicked
    }
    ...
}
```

centic



Interfaz de Usuario.

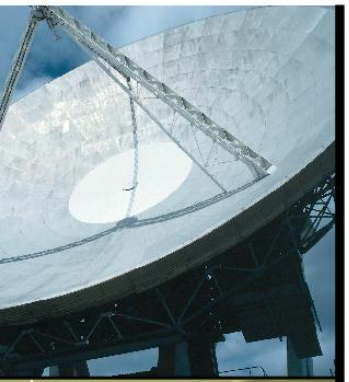
Eventos.

- Handlers
  - onKeyDown(int, KeyEvent)
    - Al pulsar una tecla
  - onKeyUp(int, KeyEvent)
    - Al levantar una pulsación
  - onTrackballEvent(MotionEvent)
    - Al tocar el pad
  - onTouchEvent(MotionEvent)
    - Al tocar la pantalla
  - onFocusChanged(boolean, int, Rect)
    - Al ganar o perder el foco

Interfaz de Usuario.

Adaptadores (AdapterView).

- Sirven para mostrar datos de cualquier fuente
- Se encargan de mostrar los datos al usuario
- Manejan las selecciones del usuario
- Necesitan un adaptador para obtener los datos
- Principales:
  - Gallery
    - Galeria de imágenes
  - ListView
    - Listado de datos
  - Spinner
    - Caja de selección múltiple.

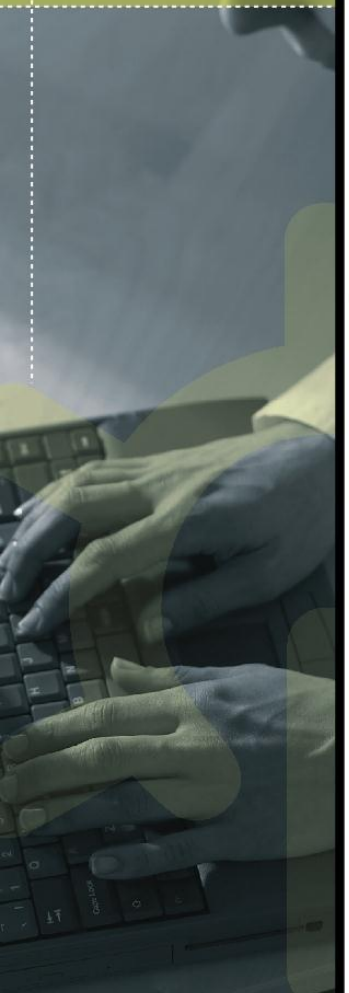


Interfaz de Usuario.

Adaptadores (AdapterView).

- Spinner
  - Layout

```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dip"
    android:text="@string/planet_prompt"
/>
<Spinner
    android:id="@+id/spinner"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:prompt="@string/planet_prompt"
/>
```

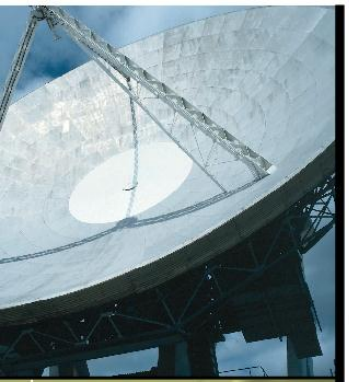


Interfaz de Usuario.

Adaptadores (AdapterView).

- Spinner
  - Values

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="planet_prompt">Choose a planet</string>
    <string-array name="planets_array">
        <item>Mercury</item>
        <item>Venus</item>
        <item>Earth</item>
        <item>Mars</item>
        <item>Jupiter</item>
        <item>Saturn</item>
        <item>Uranus</item>
        <item>Neptune</item>
    </string-array>
</resources>
```



Interfaz de Usuario.

Adaptadores (AdapterView).

- Spinner
  - Recuperar Spinner

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    Spinner spinner = (Spinner) findViewById(R.id.spinner);
    ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
            this, R.array.planets_array, android.R.layout.simple_spinner_item);
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spinner.setAdapter(adapter);
}
```

Interfaz de Usuario.

Adaptadores (AdapterView).

- Spinner
  - Registrar eventos

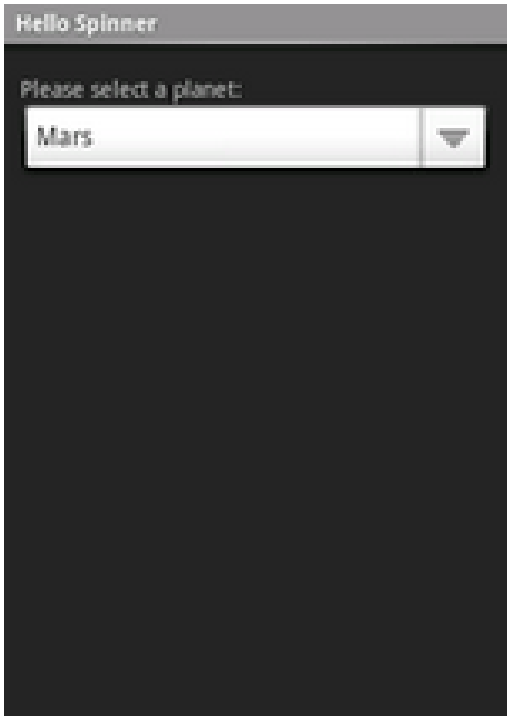
```
public class MyOnItemSelectedListener implements OnItemSelectedListener {

    public void onItemSelected(AdapterView<?> parent,
        View view, int pos, long id) {
      Toast.makeText(parent.getContext(), "The planet is " +
          parent.getItemAtPosition(pos).toString(), Toast.LENGTH_LONG).show();
    }

    public void onNothingSelected(AdapterView parent) {
      // Do nothing.
    }
}
```

```
spinner.setOnItemSelectedListener(new MyOnItemSelectedListener());
```

Interfaz de Usuario.

Adaptadores (AdapterView).

- Spinner



Interfaz de Usuario.

Adaptadores (AdapterView).

- ListView
  - Item layout

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10dp"
    android:textSize="16sp" >
</TextView>
```

centic

Interfaz de Usuario.

Adaptadores (AdapterView).

- ListView
  - Values

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="countries_array">
        <item>Bahrain</item>
        <item>Bangladesh</item>
        <item>Barbados</item>
        <item>Belarus</item>
        <item>Belgium</item>
        <item>Belize</item>
        <item>Benin</item>
    </string-array>
</resources>
```

Interfaz de Usuario.

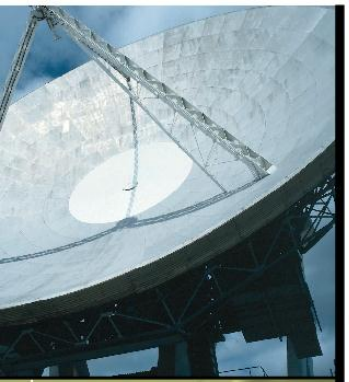
Adaptadores (AdapterView).

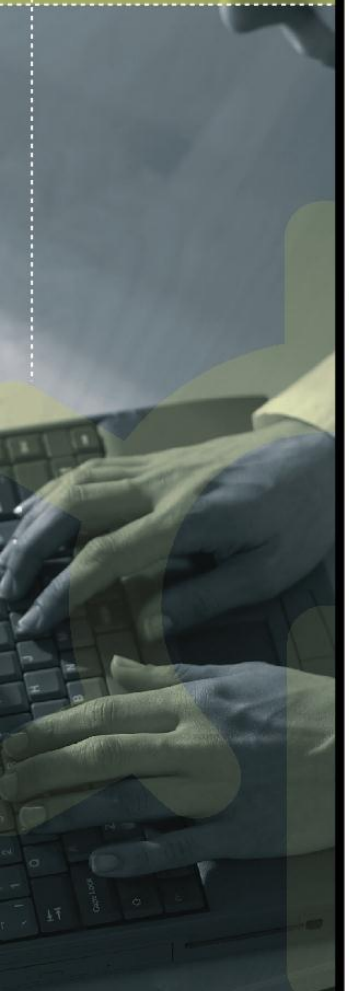
- ListView
  - Cargar datos

```
public class HelloListView extends ListActivity {
```

```
@Override
public void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);

  setListAdapter(new ArrayAdapter<String>(this, R.layout.list_item, COUNTRIES));

  ListView lv = getListView();
  lv.setTextFilterEnabled(true);

  lv.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
      // When clicked, show a toast with the TextView text
      Toast.makeText(getApplicationContext(), ((TextView) view).getText(),
          Toast.LENGTH_SHORT).show();
    }
  });
}
```

centic



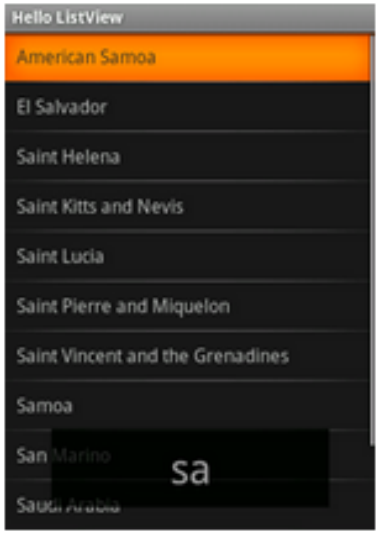


Interfaz de Usuario.

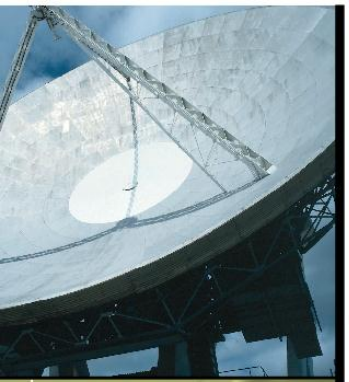
Adaptadores (AdapterView).

- ListView
  - Cargar datos

```
String[] countries = getResources().getStringArray(R.array.countries_array);
setListAdapter(new ArrayAdapter<String>(this, R.layout.list_item, countries));
```



centic



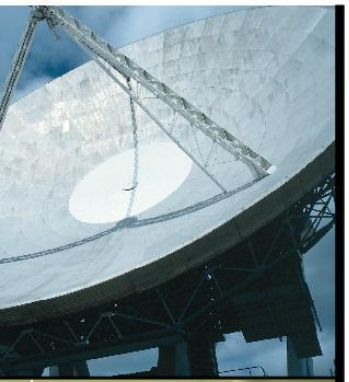
Interfaz de Usuario.

Adaptadores (AdapterView).

- GridView
  - Layout

```
<?xml version="1.0" encoding="utf-8"?>
<GridView xmlns:android="http://schemas.andro
    android:id="@+id/gridview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:columnWidth="90dp"
    android:numColumns="auto_fit"
    android:verticalSpacing="10dp"
    android:horizontalSpacing="10dp"
    android:stretchMode="columnWidth"
    android:gravity="center"
/>
```

centic



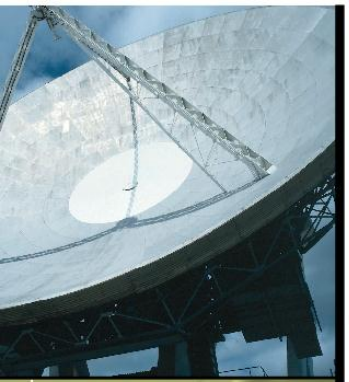
Interfaz de Usuario.

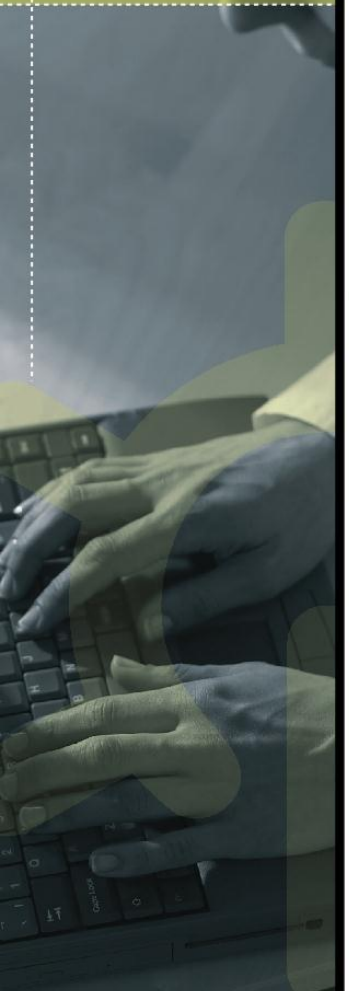
Adaptadores (AdapterView).

- GridView
  - Configurar GridView

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    GridView gridview = (GridView) findViewById(R.id.gridview);
    gridview.setAdapter(new ImageAdapter(this));
```

centic





Interfaz de Usuario.

Adaptadores (AdapterView).

- GridView
  - Registrar Eventos

```
gridview.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View v, int position, long id) {
        Toast.makeText(HelloGridView.this, "" + position, Toast.LENGTH_SHORT).show();
    }
});
```

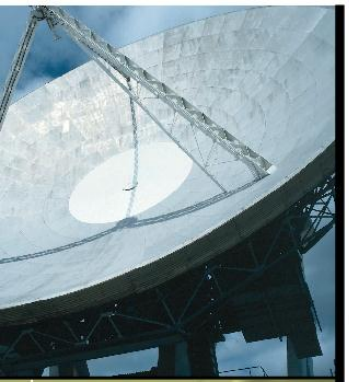
Interfaz de Usuario.

Adaptadores (AdapterView).

- GridView
  - Adaptador

```
public class ImageAdapter extends BaseAdapter {
    private Context mContext;

    public ImageAdapter(Context c) {
        mContext = c;
    }

    public int getCount() {
        return mThumbIds.length;
    }

    public Object getItem(int position) {
        return null;
    }

    public long getItemId(int position) {
        return 0;
    }
```

centic



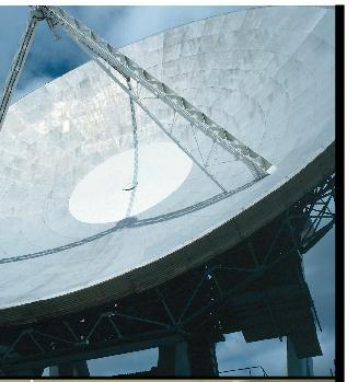
Interfaz de Usuario.

Adaptadores (AdapterView).

- GridView
  - Adaptador

```
// create a new ImageView for each item referenced by the Adapter
public View getView(int position, View convertView, ViewGroup parent) {
    ImageView imageView;
    if (convertView == null) {  // if it's not recycled, initialize some attributes
        imageView = new ImageView(mContext);
        imageView.setLayoutParams(new GridView.LayoutParams(85, 85));
        imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
        imageView.setPadding(8, 8, 8, 8);
    } else {
        imageView = (ImageView) convertView;
    }

    imageView.setImageResource(mThumbIds[position]);
    return imageView;
}
```

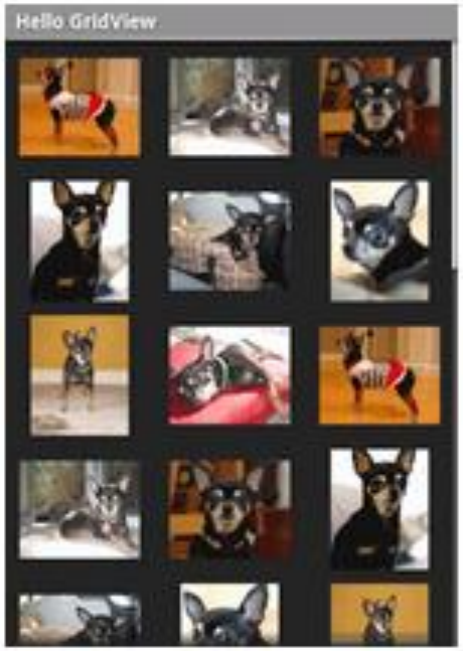
centic



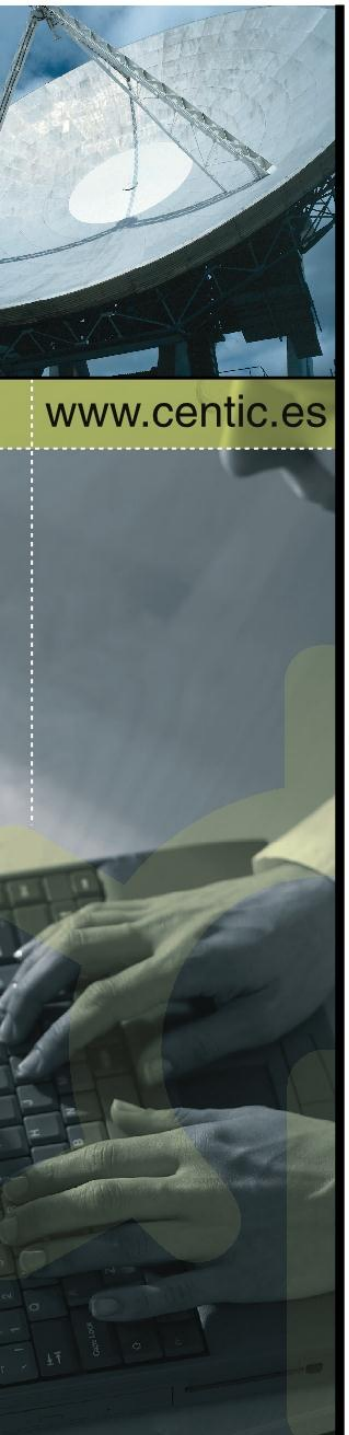
Interfaz de Usuario.

Adaptadores (AdapterView).

- GridView



centic



Interfaz de Usuario.

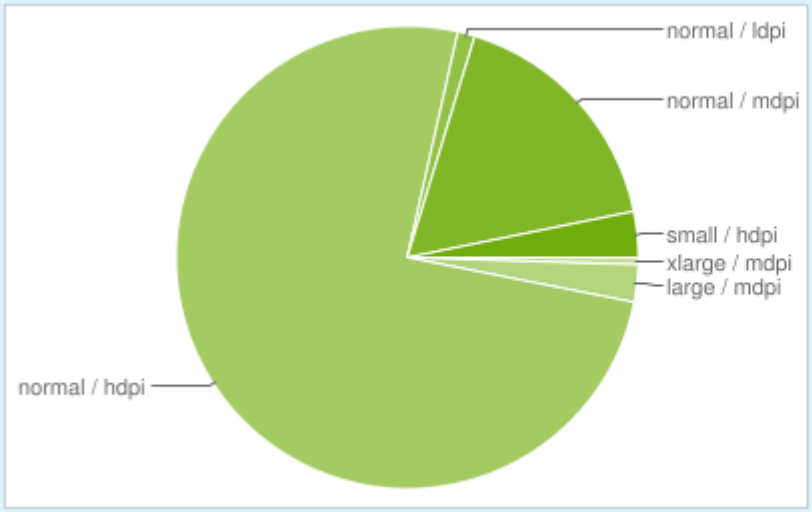
Múltiples pantallas.

- Tamaños
  - Small
  - Normal
  - Large
  - Xlarge
- Densidades
  - Ldpi
  - Mdpi
  - Hdpi
  - xhdpi

centic

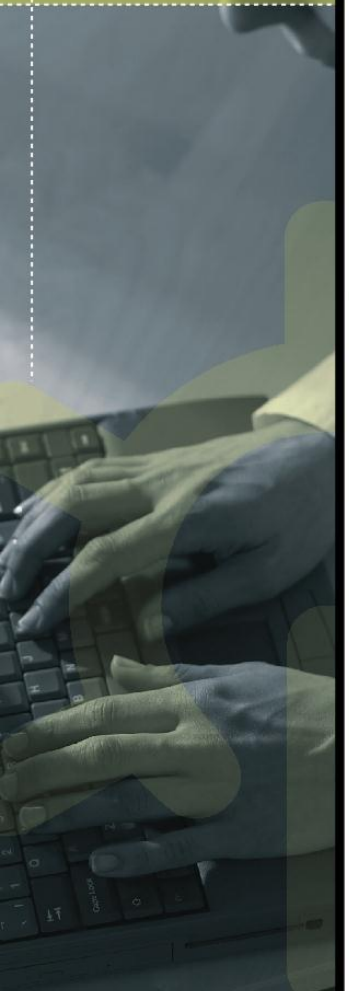
Interfaz de Usuario.

Múltiples pantallas.



	ldpi	mdpi	hdpi	xhdpi
small			3.2%	
normal	1.2%	17.1%	75.5%	
large		2.6%		
xlarge		0.5%		

centic



Interfaz de Usuario.

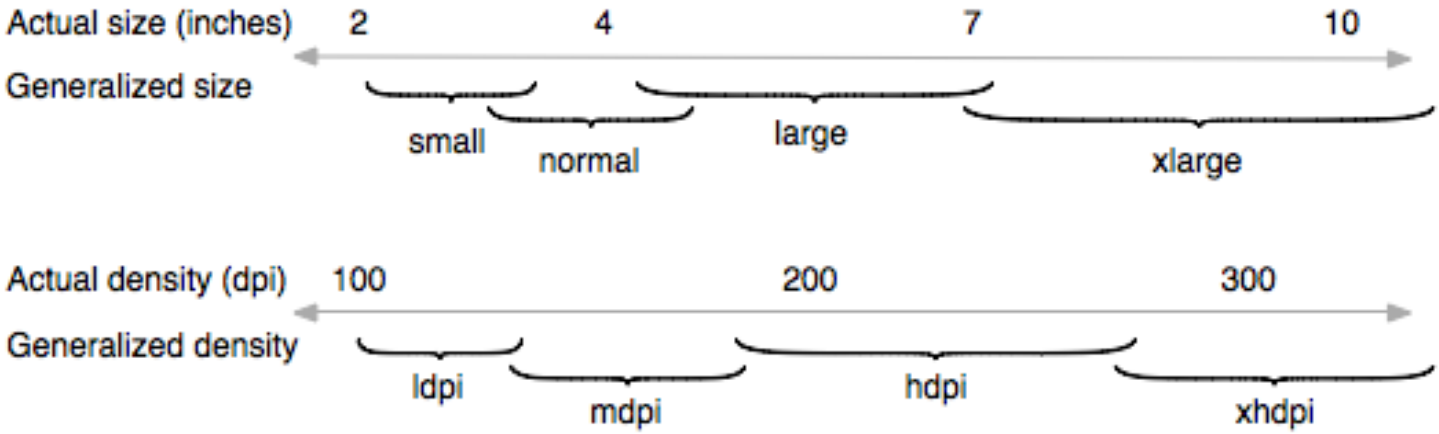
Múltiples pantallas.

- Conceptos.
  - Screen size
    - Tamaño de la pantalla
  - Screen density (dpi, dots per inch)
    - La cantidad de pixels contenidos en un área de pantalla
  - Orientation
    - Landscape
    - Portrait
  - Resolution
    - Número total de pixel de la pantalla
  - Density-independent pixel (dp)
    - Unidad de pixels virtual
    - Equivale a un pixels en 160dpi
    - px = dp * (dpi / 160)

Interfaz de Usuario.

Múltiples pantallas.

- Pantallas soportadas



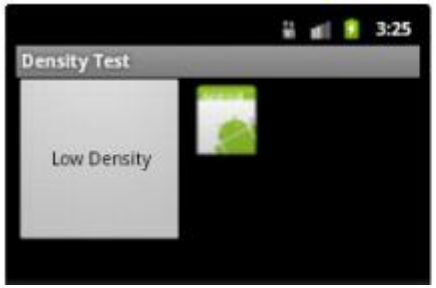
centic

Interfaz de Usuario.

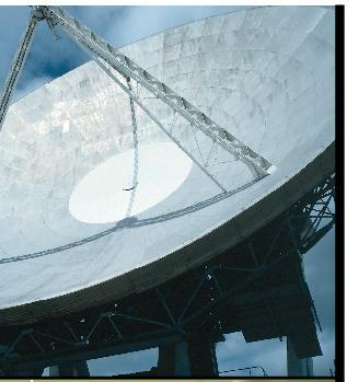
Múltiples pantallas.

- Usando Pixels

- Usando dpi



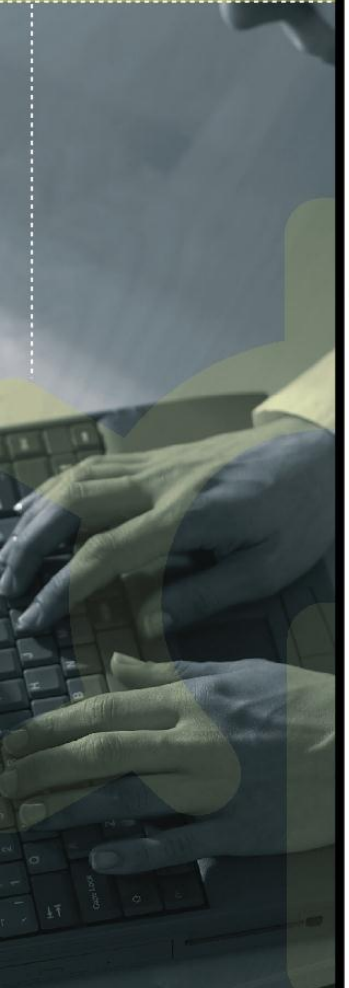
centic

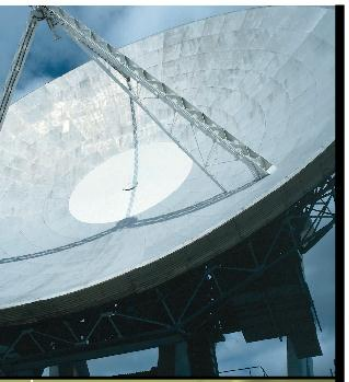


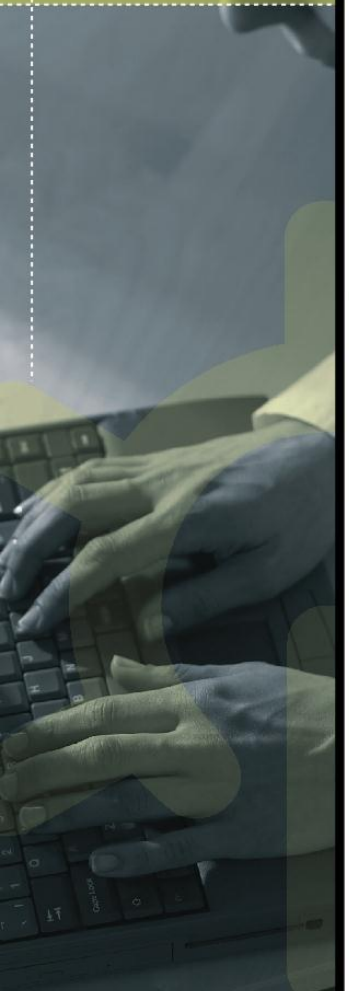
Interfaz de Usuario.

Múltiples pantallas.

- Diseñar diferentes layout para diferentes tamaños de pantalla
    - Small
    - Normal
    - Large
- Recursos
    - layout/ (Usar como predeterminado, normal)
    - layout-small/
    - layout-large/
    - layout-xlarge/







Interfaz de Usuario.

Múltiples pantallas.

- Crear imágenes para cada tipo de densidad.
- Android escala los bitmaps, si no encuentra el recurso correspondiente
  - PNG, JPG, GIF
- Recursos
  - drawable/ (Lo usamos por defecto, hdpi)
  - drawable-ldpi/
  - drawable-mdpi/

Interfaz de Usuario.

Múltiples pantallas.

- Usar combinaciones de calificadores para los recursos.

Size	`small`	Resources for *small* size screens.
	`normal`	Resources for *normal* size screens. (This is the baseline size.)
	`large`	Resources for *large* size screens.
	`xlarge`	Resources for *extra large* size screens.

centic

Interfaz de Usuario.

Múltiples pantallas.

- Usar combinaciones de calificadores para los recursos.

Density	`ldpi`	Resources for low-density (*ldpi*) screens (~120dpi).
	`mdpi`	Resources for medium-density (*mdpi*) screens (~160dpi). (T
	`hdpi`	Resources for high-density (*hdpi*) screens (~240dpi).
	`xhdpi`	Resources for extra high-density (*xhdpi*) screens (~320dpi).
	`nodpi`	Resources for all densities. These are density-independent qualifier, regardless of the current screen's density.

Interfaz de Usuario.

Múltiples pantallas.

- Usar combinaciones de calificadores para los recursos.

Orientation	`land`	Resources for screens in the landscape orientation
	`port`	Resources for screens in the portrait orientation (tal
Aspect ratio	`long`	Resources for screens that have a significantly talle than the baseline screen configuration.
	`notlong`	Resources for use screens that have an aspect ratio

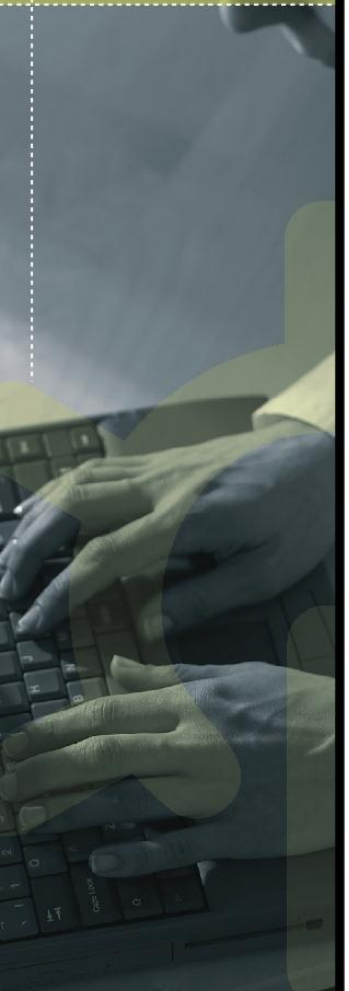
Interfaz de Usuario.

Múltiples pantallas.

- Usar combinaciones de calificadores para los recursos.
  - Ejemplos

```
res/layout/my_layout.xml
res/layout-small/my_layout.xml
res/layout-large/my_layout.xml
res/layout-xlarge/my_layout.xml
res/layout-xlarge-land/my_layout.xml

res/drawable-mdpi/my_icon.png
res/drawable-hdpi/my_icon.png
res/drawable-xhdpi/my_icon.png
```

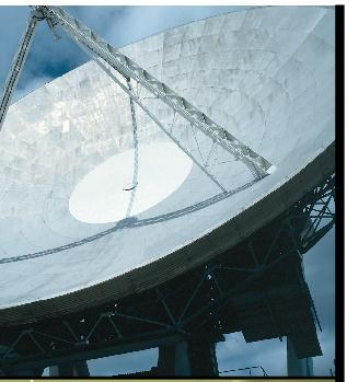
centic

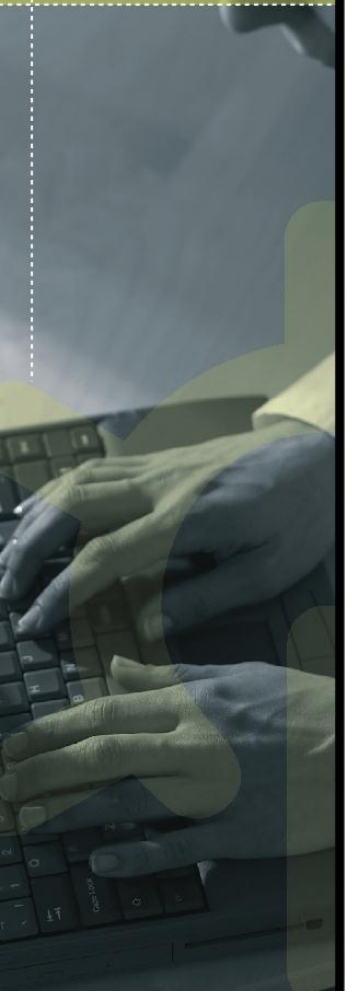


Interfaz de Usuario.

Múltiples pantallas.

- Consejos
  - Usar «wrap_content» y «fill_parent»
  - Usar «densidad independiente»
  - No usar pixels
  - No usar «AbsoluteLayout»
  - No abusar de «LinearLayout»
  - Usar «RelativeLayout» siempre que sea posible
  - Disponer de imágenes para las distintas densidades
  - Disponer de «layout» para los distintos tamaños de pantalla





Interfaz de Usuario.

Múltiples pantallas.

- Consejos
  - Usar «wrap_content» y «fill_parent»
  - Usar «densidad independiente»
  - No usar pixels
  - No usar «AbsoluteLayout»
  - No abusar de «LinearLayout»
  - Usar «RelativeLayout» siempre que sea posible
  - Disponer de imágenes para las distintas densidades
  - Disponer de «layout» para los distintos tamaños de pantalla
  - Escalar imágenes es mas costoso que tener las imágenes escaladas.
  - Crear AVD con distintas configuraciones y probar la aplicación.

centic

Interfaz de Usuario.

Múltiples pantallas.

- Resumen de pantallas y densidades

	Low density (120), *ldpi*
Small screen	QVGA (240x320)
Normal screen	WQVGA400 (240x400) WQVGA432 (240x432)
Large screen	WVGA800** (480x800) WVGA854** (480x854)
Extra Large screen	1024x600

Interfaz de Usuario.

Múltiples pantallas.

- Resumen de pantallas y densidades

	Medium density (160), *mdpi*
Small screen	
Normal screen	HVGA (320x480)
Large screen	WVGA800* (480x800) WVGA854* (480x854) 600x1024
Extra Large screen	WXGA (1280x800)† 1024x768 1280x768

centic

Interfaz de Usuario.

Múltiples pantallas.

- Resumen de pantallas y densidades

	High density (240), *hdpi*
Small screen	480x640
Normal screen	WVGA800 (480x800) WVGA854 (480x854) 600x1024
Large screen	
Extra Large screen	1536x1152 1920x1152 1920x1200

centic

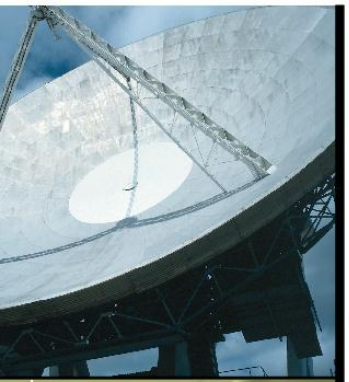
Interfaz de Usuario.

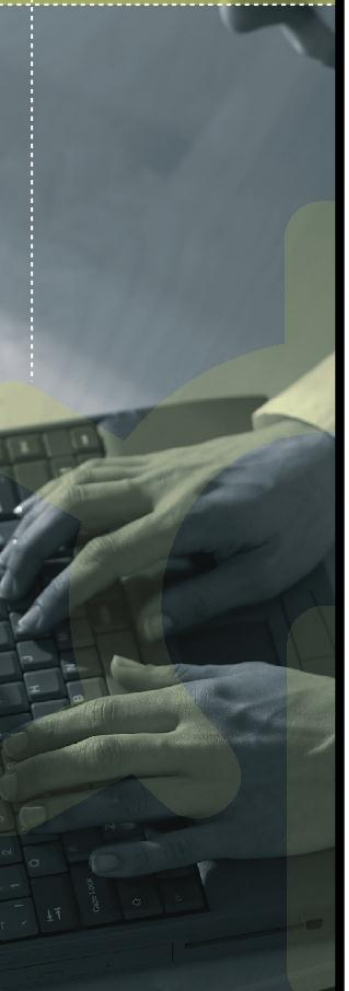
Múltiples pantallas.

- Resumen de pantallas y densidades

	Extra high density (320), *xhdpi*
Small **screen**	
Normal **screen**	640x960
Large **screen**	
Extra Large **screen**	2048x1536 2560x1536 2560x1600

centic





Interfaz de Usuario.

Múltiples pantallas.

- Diseño de iconos
    - Plantillas de iconos

http://developer.android.com/shareables/icon_templates-v2.3.zip

http://developer.android.com/shareables/icon_templates-v2.0.zip

http://developer.android.com/shareables/icon_templates-v1.0.zip

Interfaz de Usuario.

Múltiples pantallas.

- Diseño de iconos
  - Tamaños

Icon Type	Standard Asset Sizes (in F
	Low density screen *(ldpi)*
Launcher	36 x 36 px
Menu	36 x 36 px
Status Bar (Android 2.3 and later)	12w x 19h px (preferred, width may vary)
Status Bar (Android 2.2 and below)	19 x 19 px
Tab	24 x 24 px
Dialog	24 x 24 px
List View	24 x 24 px

Interfaz de Usuario.

Múltiples pantallas.

- Diseño de iconos
  - Tamaños

Icon Type	Pixels), for Generalized Screen D
	Medium density screen *(mdpi)*
Launcher	48 x 48 px
Menu	48 x 48 px
Status Bar (Android 2.3 and later)	16w x 25h px (preferred, width may vary)
Status Bar (Android 2.2 and below)	25 x 25 px
Tab	32 x 32 px
Dialog	32 x 32 px
List View	32 x 32 px

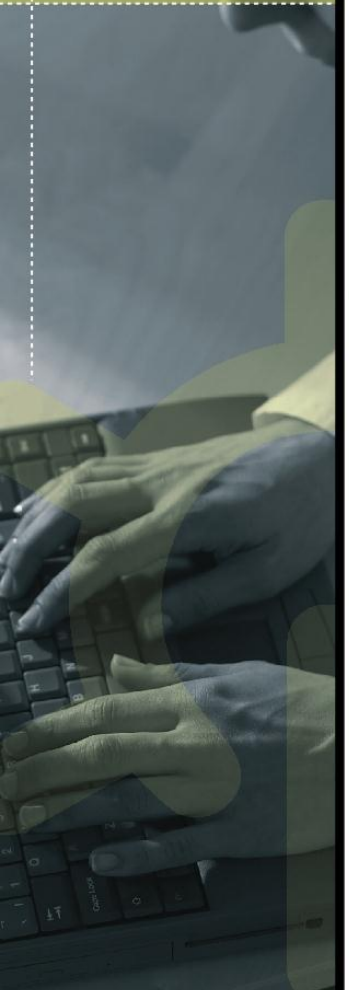
centic

Interfaz de Usuario.

Múltiples pantallas.

- Diseño de iconos
  - Tamaños

Icon Type	Densities
	High density screen *(hdpi)*
Launcher	72 x 72 px
Menu	72 x 72 px
Status Bar (Android 2.3 and later)	24w x 38h px (preferred, width may vary)
Status Bar (Android 2.2 and below)	38 x 38 px
Tab	48 x 48 px
Dialog	48 x 48 px
List View	48 x 48 px



Interfaz de Usuario.

Ejercicios

- Implementar una calculadora básica

- Implementar una galería de fotos
  - Listado de fotos
  - Ver una foto