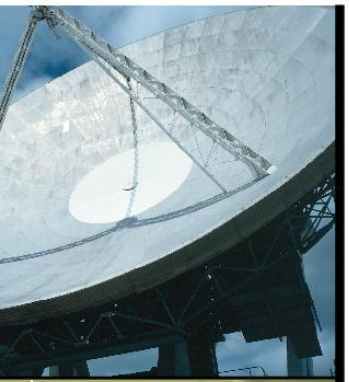
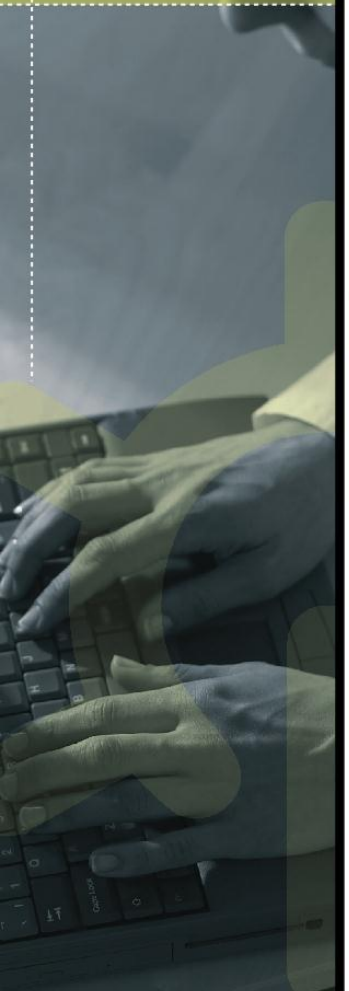
www.centic.es

Curso Android
Edición **2011**

CENTRO TECNOLÓGICO DE LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

centic

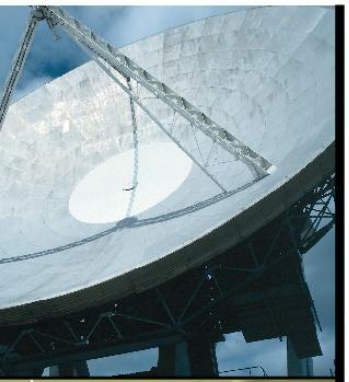




Intents.

Funcionalidad

- Son mensajes que indican acciones a ejecutar
- Llaman a Activities
  - Comparten datos con otras Activities
  - startActivity()
- Inician Servicios
  - startService()
- Envían Broadcast
  - sendBroadcast()



Intents.

Estructura

- Component Name
  - El nombre del componente que debe realizar la acción
  - «ComponentName» es opcional
  - Si no está declarado se usa el resto de información para seleccionar el componente que debe ejecutar la acción

```
Intent it = new Intent(CursoAndroid.this, ActivityALanzar.class);
startActivity(it);
```

centic

Intents.

Estructura

- Action
  - String con la acción a realizar

Constant	Target component
ACTION_CALL	activity
ACTION_EDIT	activity
ACTION_MAIN	activity
ACTION_SYNC	activity
ACTION_BATTERY_LOW	broadcast receiver
ACTION_HEADSET_PLUG	broadcast receiver
ACTION_SCREEN_ON	broadcast receiver
ACTION_TIMEZONE_CHANGED	broadcast receiver

Intents.

Estructura

- Data
  - Es opcional
  - Distintos tipos de acciones requieren distintos tipos de datos
  - Indicamos su «MIME type»
  - Ejemplos
    - ACTION_CALL: tel:555555555
    - ACTION_VIEW: http://google.com

www.centic.es

Intents.

Estructura

- Category
  - String con información adicional sobre el componente que debe gestionar la acción

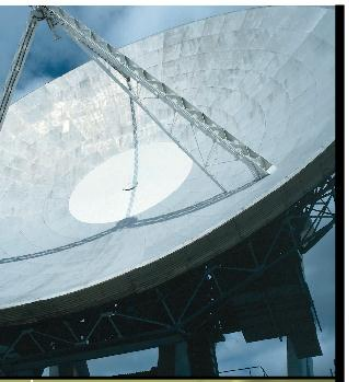
Constant
CATEGORY_BROWSABLE
CATEGORY_GADGET
CATEGORY_HOME
CATEGORY_LAUNCHER
CATEGORY_PREFERENCE

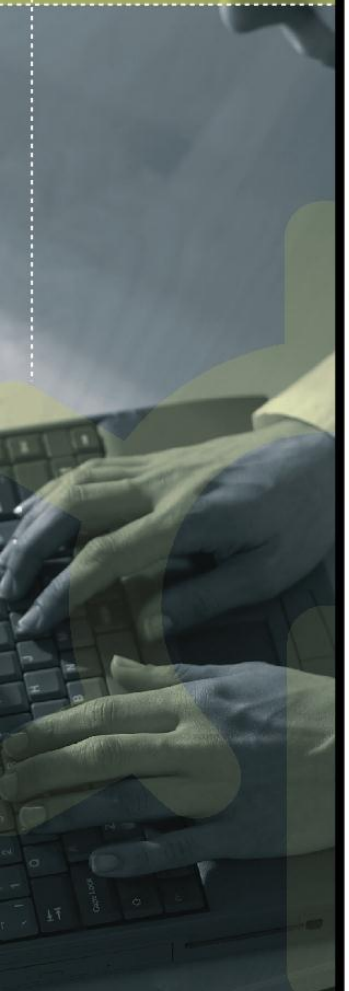
centic

Intents.

Estructura

- Extras
  - Usa el objeto Bundle
  - Almacena la información que queremos enviar.
  - Se almacenan en un diccionario (Key:Value)
  - putXXX()
  - getXXX()
  - putExtras()
  - getExtras()
  - Existen claves predefinidas para algunos tipos de datos
- Flags
  - Modifican el comportamiento del componente destino





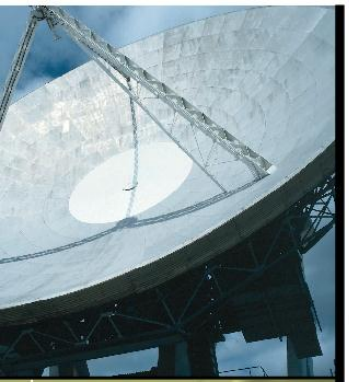
Intents.

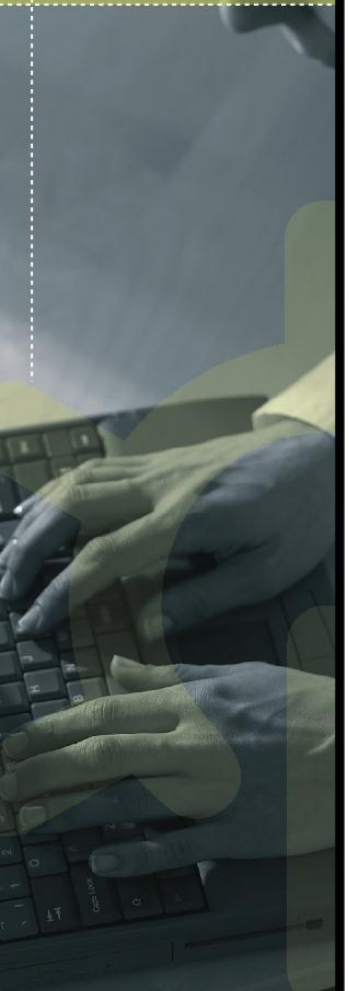
Gestión de Intents.

- Explícitos
  - Designados por su ComponentName
  - Se entregan al componente designado directamente

- Implícitos
  - No tienen «ComponentName»
  - Se usan para llamar a componentes de otras aplicaciones
  - Android selecciona el componente adecuado para el «Intent»
  - Selección basada en estructuras «Intent-Filters»
  - Datos comparados
    - ACTION
    - DATA
    - CATEGORY

centic





Intents.

Gestión de Intents.

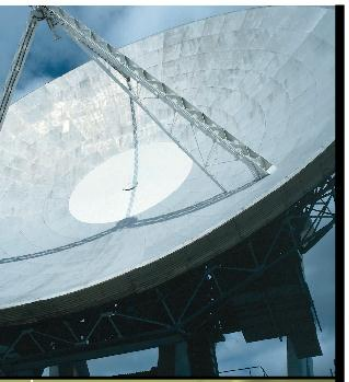
- Intent-Filter
  - Representan las capacidades de los componentes
    - Activity
    - Service
    - Broadcast
  - Cada componente puede tener de 0 a N
  - Se usa uno por cada funcionalidad del componente
  - Se publican en el «Manifest.xml»
  - Cada «Intent-Filter» declarado es añadido al listado del sistema
  - Cada «Intent» enviado se comprueba contra el listado de «Intent-Filter» del sistema

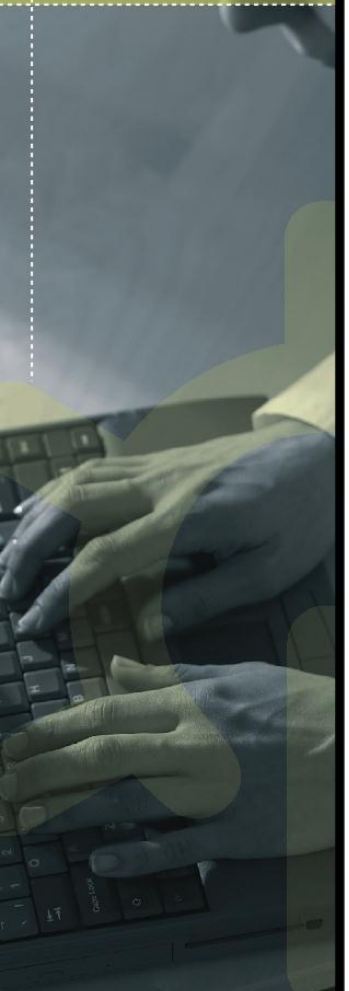
Intents.

Gestión de Intents.

- Proceso de filtrado
  - Cada «Intent» debe pasar por 3 fases de filtrado
  - Comprobar su «ACTION»
  - Comprobar su «CATEGORY»
  - Comprobar su «DATA»

```
<intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <action android:name="android.intent.action.EDIT" />
    <action android:name="android.intent.action.PICK" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:mimeType="vnd.android.cursor.dir/vnd.google.note" />
</intent-filter>
```

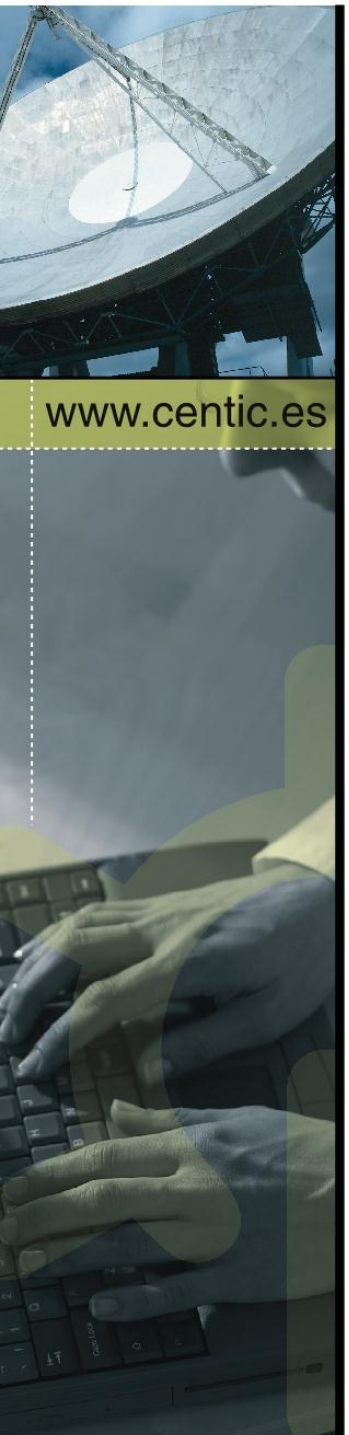




Intents.

Gestión de Intents.

- Proceso de filtrado
  - ACTION
    - Un filtro debe contener al menos una acción
    - Para pasar el test la acción del «Intent» debe estar contenida en el filtro
    - Si el «Intent» especifica acción y esta no se encuentra en el filtro, este es descartado
    - Si el «Intent» no especifica acción el filtro es candidato
    - Los filtros candidatos pasan al siguiente test



Intents.

Gestión de Intents.

- Proceso de filtrado
  - CATEGORY
    - Todas las categorías de un «Intent» deben estar reflejadas en el filtro
    - Si el «Intent» incluye una categoría que no esta en el filtro, este es descartado
    - El filtro puede incluir mas categorías que el «Intent»
    - Un Intent sin categorías «siempre» pasa el filtro
    - Un filtro sin categorías requiere un «Intent» sin categorías
    - startActivity() siempre añade la categoría DEFAULT a sus «Intents»

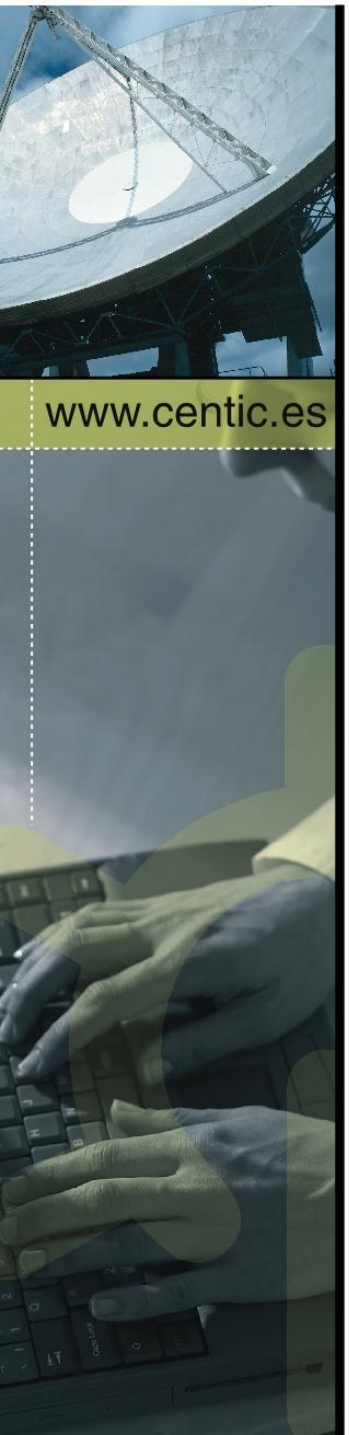
centic

Intents.

Gestión de Intents.

- Proceso de filtrado
  - DATA
    - Pueden aparecer de 0 a N elementos DATA
    - Uri scheme://host:port/path
      - scheme
      - host
      - port (sin host se ignora)
      - path
      - Atributos opcionales pero dependientes entre ellos
      - authority = host + port
  - type especifica el «MIME type»

centic

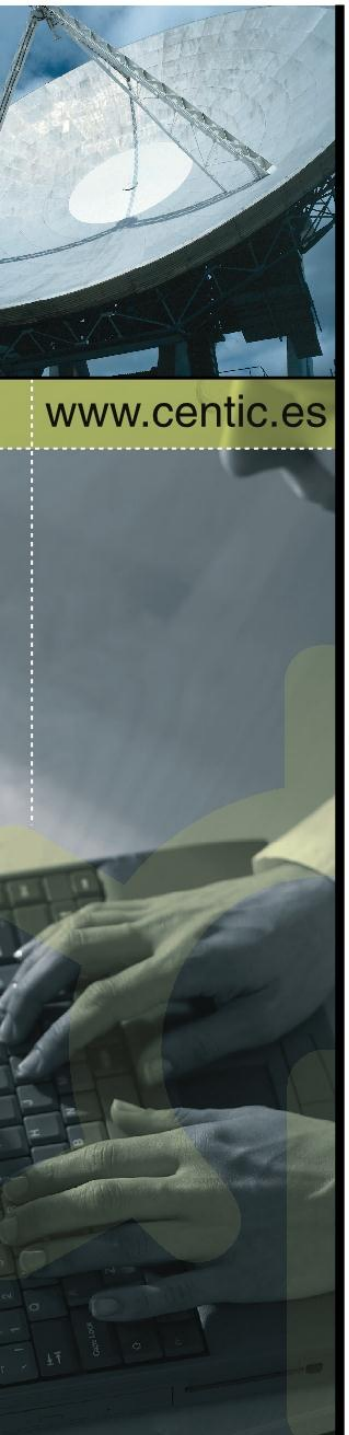


Intents.

Gestión de Intents.

- Proceso de filtrado
  - DATA
    - Sólo se comparan los atributos especificados en el filtro
    - El «Intent» debe contener todos los atributos que se especifiquen en el filtro y estos deben coincidir
    - El atributo «path» y «type» pueden usar comodines «*»
      - content://contacts/*
      - content://contacts/23
      - text/*
      - image/*

centic



Intents.

Gestión de Intents.

- Proceso de filtrado
  - DATA
    - Un «Intent» que no contiene «data» sólo pasa el test si el filtro no contiene «data»
    - Un «Intent» que solo contiene URI pasa el test sólo si el filtro solo contiene Uri y estas coinciden
    - Un «Intent» que solo contiene «type» pasa el test sólo si el filtro solo contiene «type» y estos coinciden o al menos el del filtro es menos restrictivo
    - Un «Intent» que contiene Uri y Type pasa el test solo si coinciden su type y su Uri

centic

Intents.

Gestión de Intents.

- Buscando Intents

```
public static boolean isIntentAvailable(Context context, String action) {
    final PackageManager packageManager = context.getPackageManager();
    final Intent intent = new Intent(action);
    List<ResolveInfo> list =
            packageManager.queryIntentActivities(intent,
                    PackageManager.MATCH_DEFAULT_ONLY);
    return list.size() > 0;
}
```

Intents.

Gestión de Intents.

- Buscando Intents

```
@Override
public boolean onPrepareOptionsMenu(Menu menu) {
    final boolean scanAvailable = isIntentAvailable(this,
        "com.google.zxing.client.android.SCAN");

    MenuItem item;
    item = menu.findItem(R.id.menu_item_add);
    item.setEnabled(scanAvailable);

    return super.onPrepareOptionsMenu(menu);
}
```

centic

Intents.

Ejemplo de Intents.

```
public void takePhoto(View view){
    Intent intent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE );
    startActivityForResult(intent, RESPONSE_TAKE_PHOTO);
}
```

```
public void shareLink(View v){
    Intent intent = new Intent(Intent.ACTION_SEND);
    intent.setType("text/plain");
    intent.putExtra(Intent.EXTRA_TEXT, "Curso de Android en el centic!!! http://centic.es");
    startActivity(Intent.createChooser(intent, "Compartir con:"));
}
```

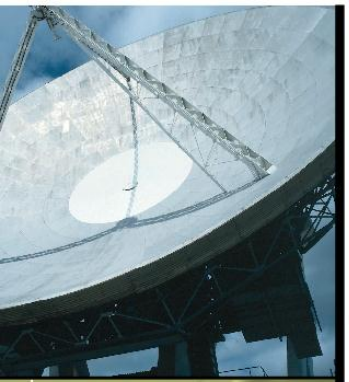
centic

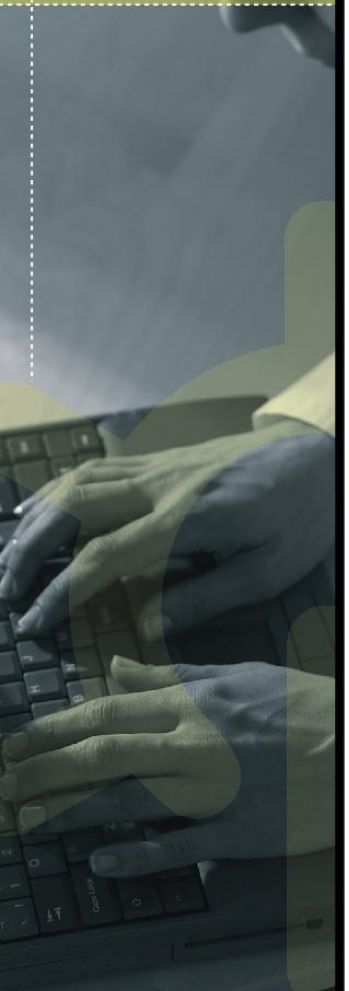
Intents.

Ejemplo de Intents.

```
public void showWeb(View v){
    Intent intent = new Intent(Intent.ACTION_VIEW);
    intent.setData(Uri.parse("http://centic.es"));
    startActivity(Intent.createChooser(intent, "Compartir con:"));
}
```

```
public void makeCall(View v){
    Intent intent = new Intent(Intent.ACTION_DIAL);
    intent.setData(Uri.parse("tel:123456789"));
}
```





Menús.

Tipos de Menús.

- Options Menu
  - La colección principal de los elementos de menú para una Activity
  - Aparece cuando el usuario toca el botón MENU
- Context Menu
  - Una lista flotante de los elementos del menú
  - Aparece cuando el usuario mantiene pulsado un componente
- Submenu
  - Una lista flotante de los elementos del menú
  - Aparece cuando el usuario toca un elemento de menú que contiene un menú anidado

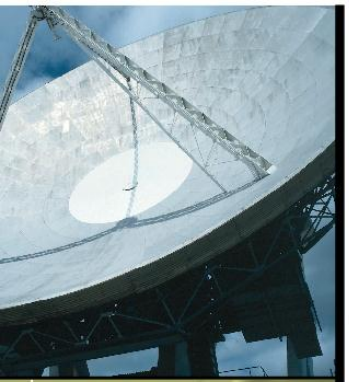
www.centic.es

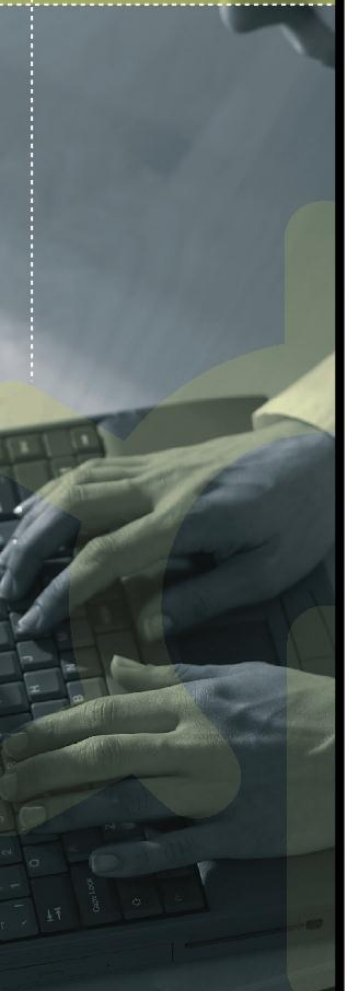
Menús.

Crear Menús.

- Debe definirse el menú en XML
  - /res/menu/menu_sample.xml
- Elementos
- <menu>
  - Es el elemento contenedor para los elementos del menú
  - Debe ser un nodo raiz
- <item>
  - Crea un MenuItem
  - Representa cada elemento del menú
  - Puede anidar a otros <menu>

centic



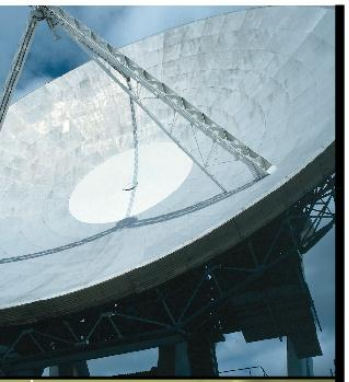


Menús.

Crear Menús.

- <group>
  - Permite englobar a los elementos <menu> que comparten propiedades

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/new_game"
          android:icon="@drawable/ic_new_game"
          android:title="@string/new_game" />
    <item android:id="@+id/help"
          android:icon="@drawable/ic_help"
          android:title="@string/help" />
</menu>
```



www.centic.es

Menús.

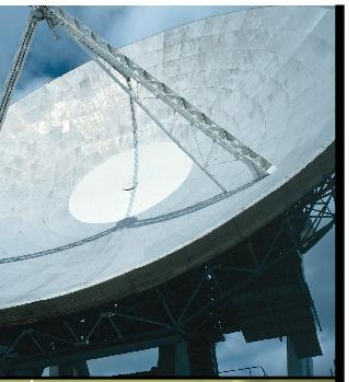
Crear Menús.

- android:id
  - Un identificador de recurso que es único en el menú
  - Permite que la aplicación puede reconocer el elemento cuando el usuario lo selecciona

- android:icon
  - Una referencia al recurso a utilizar como icono del elemento

- android:title
  - Una referencia a una cadena que se utiliza como título del artículo

centic



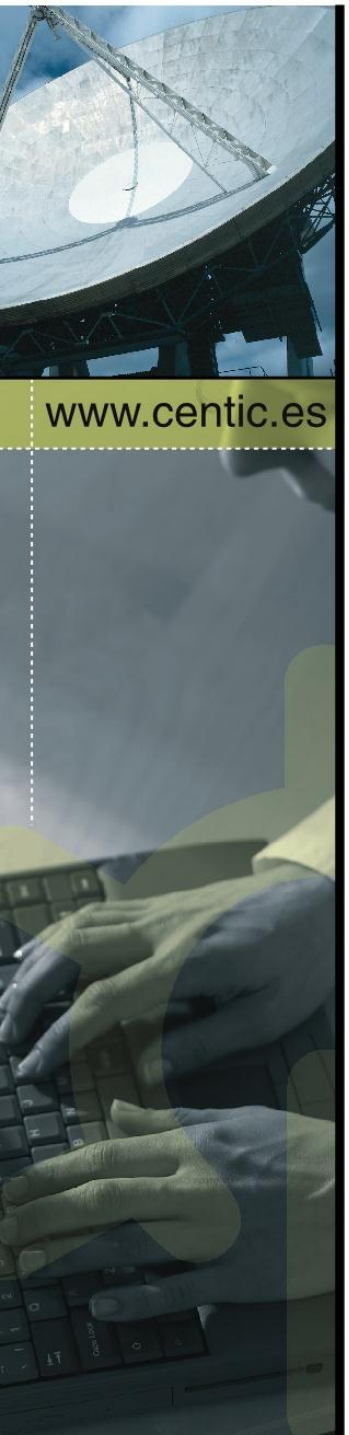
Menús.

Cargar Menús.

- Convertir los recursos de XML en un objeto
- Máximo 6 elementos
- Genera submenú automático
- getMenuInflater()
- MenuInflater.inflate()
- onCreateOptionsMenu()

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

centic



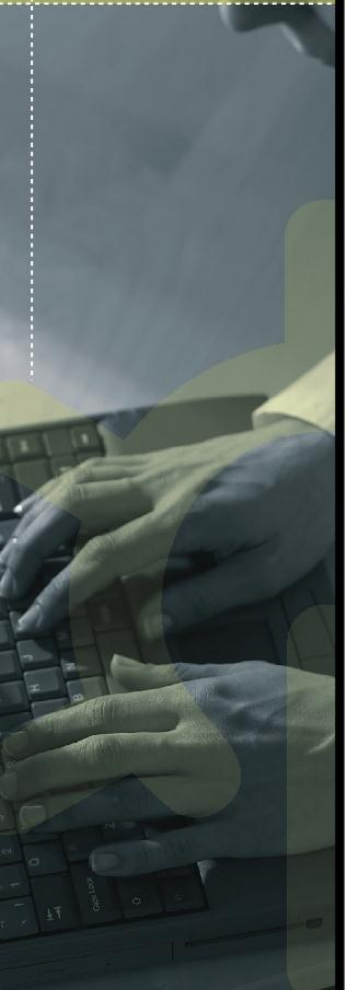
Menús.

Responder a Menús.

- onOptionsItemSelected()
- getItemId() ⇔ android:id

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle item selection
    switch (item.getItemId()) {
    case R.id.new_game:
        newGame();
        return true;
    case R.id.help:
        showHelp();
        return true;
    default:
        return super.onOptionsItemSelected(item);
    }
}
```

centic

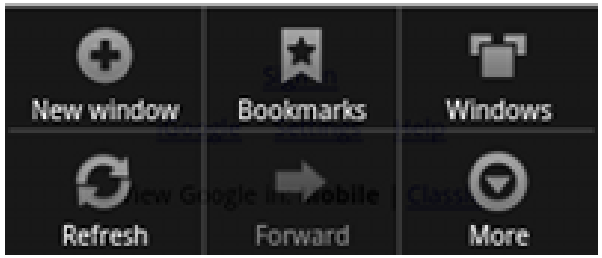


Menús.

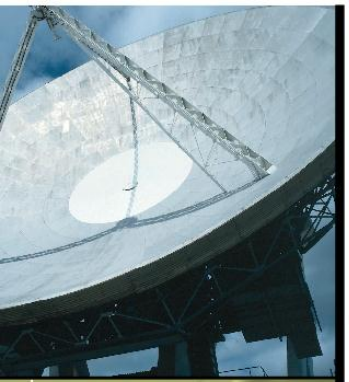
Más sobre Menús.

- Menús en todas las Activities
  - Crear Activity con onCreateOptionsMenu()
  - Implementar onOptionsItemSelected()
  - Heredar el resto de Activities

- Cambios en el menú
  - Implementar onPrepareOptionsMenu()



centic



www.centic.es

Menús.

Menú Contextual.

- Es conceptualmente similar al menú que aparece cuando el usuario realiza un "botón derecho del ratón" en un PC
- Podemos añadirlos a cualquier componente
- Se activan con una pulsación larga sobre el componente
- El componente debe registrar el menú
  - registerForContextMenu(View v)

```
EditText et = new EditText(this);
registerForContextMenu(et);
```

centic

Menús.

Menú Contextual.

- Cargar el menú

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
                                ContextMenuInfo menuInfo) {
  super.onCreateContextMenu(menu, v, menuInfo);
  MenuInflater inflater = getMenuInflater();
  inflater.inflate(R.menu.context_menu, menu);
}
```

centic

Menús.

Menú Contextual.

- Responder al menú

```
@Override
public boolean onContextItemSelected(MenuItem item) {
  AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();
  switch (item.getItemId()) {
  case R.id.edit:
    editNote(info.id);
    return true;
  case R.id.delete:
    deleteNote(info.id);
    return true;
  default:
    return super.onContextItemSelected(item);
  }
}
```

Menús.

Submenú.

- Podemos añadir un submenú a cualquier menú
- No podemos anidar submenús

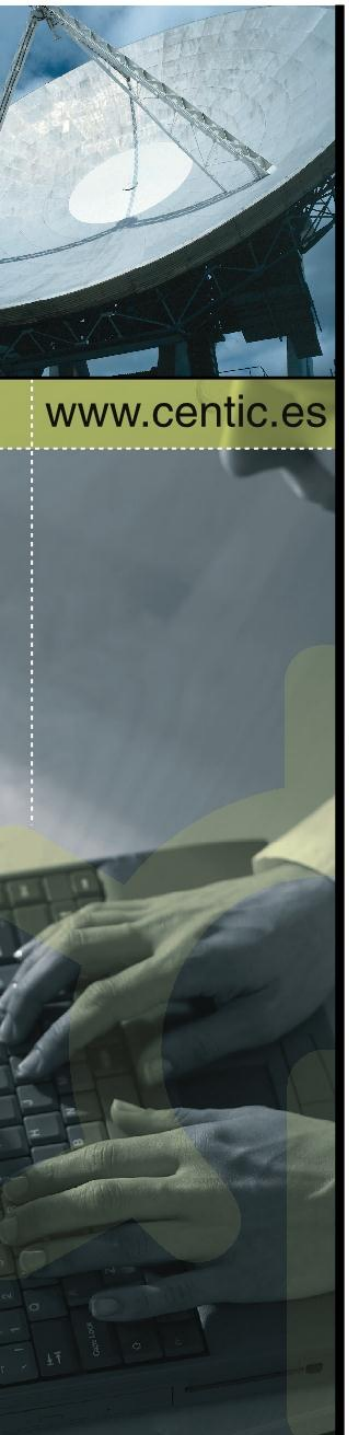
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/file"
          android:icon="@drawable/file"
          android:title="@string/file" >
        <!-- "file" submenu -->
        <menu>
            <item android:id="@+id/create_new"
                  android:title="@string/create_new" />
            <item android:id="@+id/open"
                  android:title="@string/open" />
        </menu>
    </item>
</menu>
```

centic

Menús.

Consejos

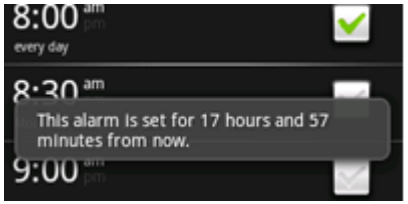
- Separar los comandos de selección de los comandos de acción
- Ordene los menús según la frecuencia de utilización de sus elementos
- No poner los comandos sólo en el menú contextual
- Organizar los menús contextuales en función del componente que los registra
- Utilizar nombres cortos y descriptivos
- No se deben poner menús a los dialogos
- Ocultar las acciones que no están disponibles.



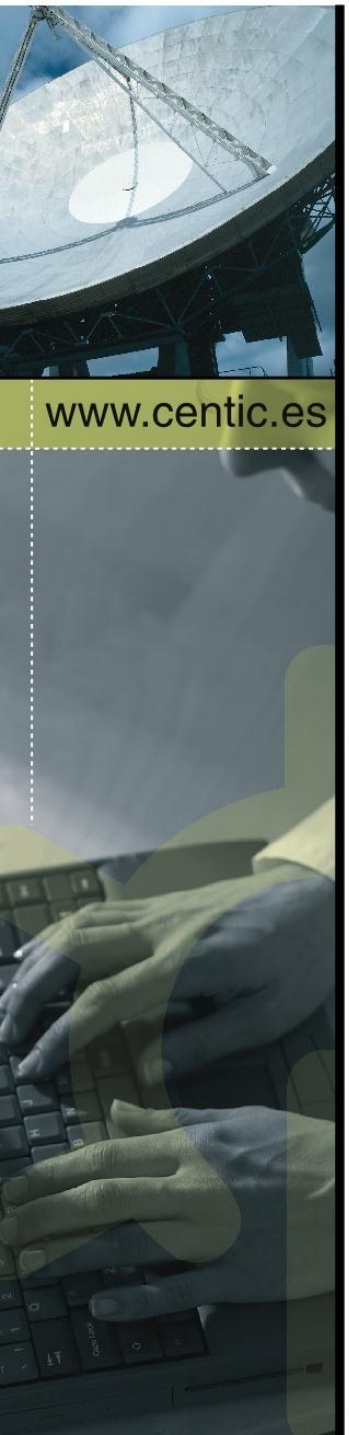
Notificaciones.

Tipos de Notificaciones.

- Toast Notification
  - Un Toast es un mensaje que aparece en la superficie de la ventana.
  - Sólo usa la cantidad de espacio requerido para el mensaje
  - La notificación automáticamente se desvanece
  - Un Toast es ideal para mensajes de texto cortos
  - Se usa cuando estás bastante seguro de que el usuario está prestando atención a la pantalla
  - Un toast no puede aceptar la interacción eventos de usuario



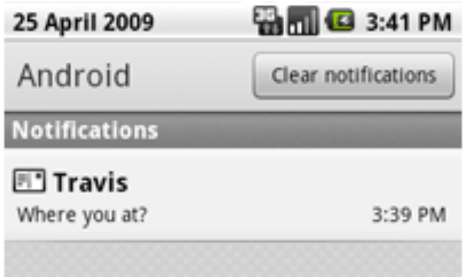
centic



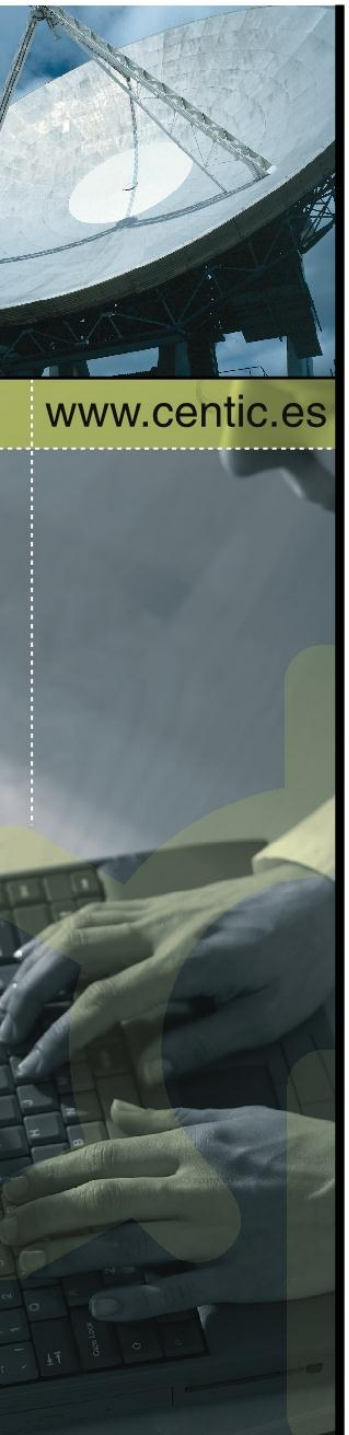
Notificaciones.

Tipos de Notificaciones.

- Status Bar Notification
  - La notificación de la barra de estado añade un icono a la barra de estado del sistema y un mensaje ampliado en la sección "Notificaciones" de la ventana.
  - Cuando el usuario selecciona el mensaje ampliado, Android lanza el «Intent» asociado a la notificación
  - También podemos configurar la notificación para alertar al usuario con un sonido, una vibración y luces intermitentes en el dispositivo.
  - Este tipo de notificación es ideal cuando trabajamos con servicios o hilos



centic



Notificaciones.

Tipos de Notificaciones.

- Diálogos
  - Un diálogo es una pequeña ventana que aparece al frente de la actividad actual
  - La actividad subyacente pierde el foco y el cuadro de diálogo acepta toda la interacción del usuario
  - Los cuadros de diálogo se utilizan normalmente para las notificaciones y actividades cortas que se relacionan directamente con la aplicación en curso.
  - Debemos usar un cuadro de diálogo cuando queramos mostrar una barra de progreso o un mensaje corto que requiere la confirmación del usuario
  - Podemos utilizar también los cuadros de diálogo como componentes integrados en nuestra aplicación

centic

Notificaciones.

Toast.

- Elementos
  - Contexto
  - Mensaje
  - Duración

```
Context context = getApplicationContext();
CharSequence text = "Hello toast!";
int duration = Toast.LENGTH_SHORT;

Toast toast = Toast.makeText(context, text, duration);
toast.show();
```

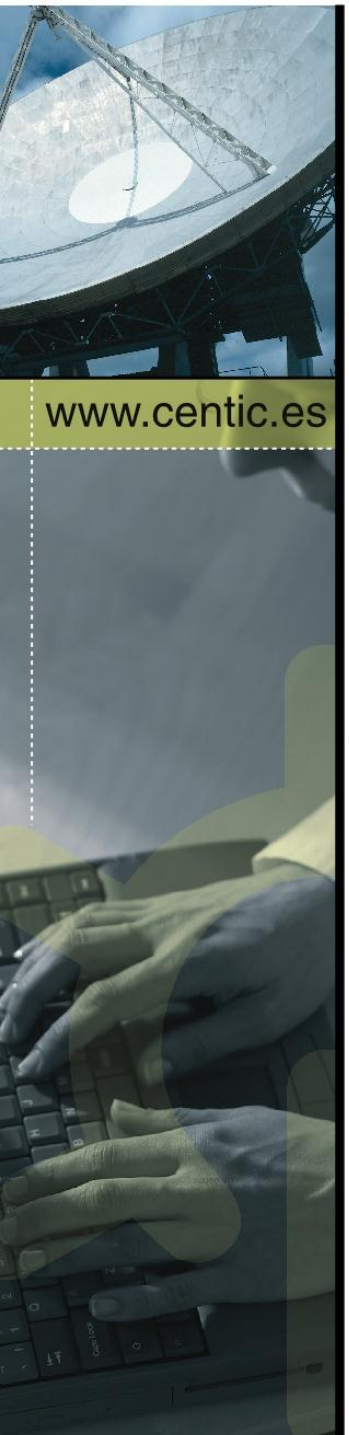
centic

Notificaciones.

Toast.

- Posicionamiento

```
Toast toast = Toast.makeText(this, "Hola Mundo!!!", Toast.LENGTH_LONG);

toast.setGravity(Gravity.CENTER_HORIZONTAL, 10, 10);
toast.setGravity(Gravity.LEFT | Gravity.TOP, 15, 15);
toast.setGravity(Gravity.CENTER_HORIZONTAL, 0, 0);
toast.setGravity(Gravity.RIGHT, 20, 0);
```



Notificaciones.

Toast.

- Personalizados
  - Crear Layout

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:id="@+id/toast_layout_root"
              android:orientation="horizontal"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent"
              android:padding="10dp"
              android:background="#DAAA"
              >
    <ImageView android:id="@+id/image"
               android:layout_width="wrap_content"
               android:layout_height="fill_parent"
               android:layout_marginRight="10dp"
               />
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="fill_parent"
              android:textColor="#FFF"
              />
</LinearLayout>
```

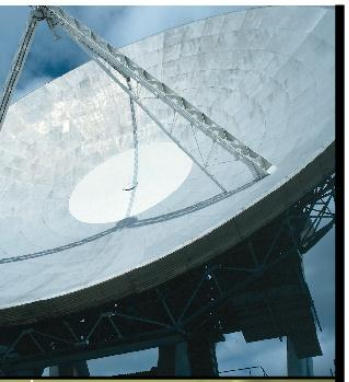
Notificaciones.

Toast.

- Personalizados
  - Cargar Layout

```
LayoutInflater inflater = getLayoutInflater();
View layout = inflater.inflate(R.layout.toast_layout,
                               (ViewGroup) findViewById(R.id.toast_layout_root));

ImageView image = (ImageView) layout.findViewById(R.id.image);
image.setImageResource(R.drawable.android);
TextView text = (TextView) layout.findViewById(R.id.text);
text.setText("Hello! This is a custom toast!");

Toast toast = new Toast(getApplicationContext());
toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);
toast.setDuration(Toast.LENGTH_LONG);
toast.setView(layout);
toast.show();
```

centic



Notificaciones.

Barra de Notificaciones.

- Obtenemos el gestor de notificaciones

```
String ns = Context.NOTIFICATION_SERVICE;
NotificationManager mNotificationManager = (NotificationManager) getSystemService(ns);
```

- Instanciar una notificación

```
int icon = R.drawable.notification_icon;
CharSequence tickerText = "Hello";
long when = System.currentTimeMillis();

Notification notification = new Notification(icon, tickerText, when);
```

Notificaciones.

Barra de Notificaciones.

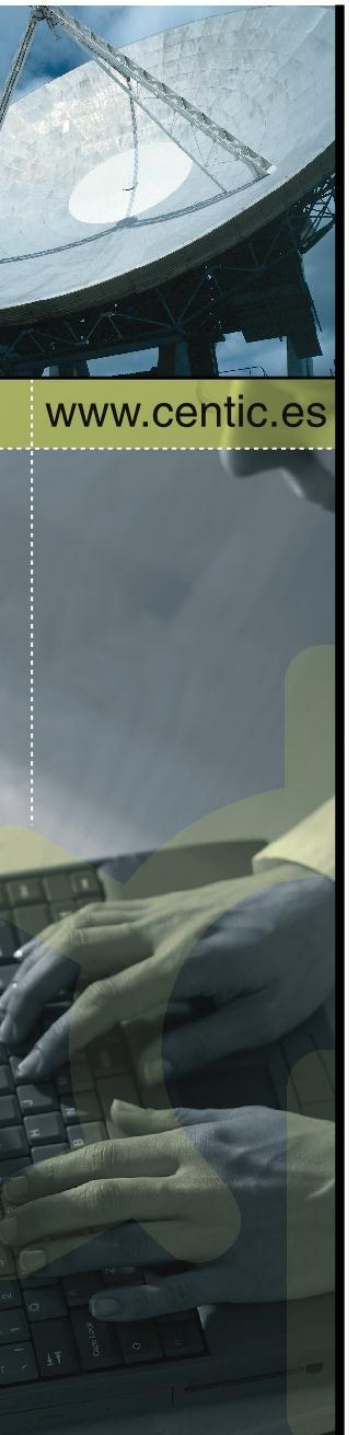
- Definir el mensaje expandido de la notificación

```
Context context = getApplicationContext();
CharSequence contentTitle = "My notification";
CharSequence contentText = "Hello World!";
Intent notificationIntent = new Intent(this, MyClass.class);
PendingIntent contentIntent = PendingIntent.getActivity(this, 0, notificationIntent, 0);

notification.setLatestEventInfo(context, contentTitle, contentText, contentIntent);
```

- Lanzar la notificación

```
private static final int HELLO_ID = 1;

mNotificationManager.notify(HELLO_ID, notification);
```

centic



Notificaciones.

Barra de Notificaciones.

- Crear una notificación
  - Elementos necesarios
    - Un icono de la barra de estado
    - Un mensaje de titulo
    - Un mensaje expandido
    - Un «PendingIntent» que se activa cuando la notificación se selecciona
  - Elementos opcionales
    - Un mensaje de texto para la barra de estado
    - Un sonido de alerta
    - Un patrón para vibrar
    - Un patrón para parpadear

centic

Notificaciones.

Barra de Notificaciones.

- Crear una notificación

```
int icon = R.drawable.notification_icon;        // icon from resources
CharSequence tickerText = "Hello";              // ticker-text
long when = System.currentTimeMillis();         // notification time
Context context = getApplicationContext();      // application Context
CharSequence contentTitle = "My notification";  // expanded message title
CharSequence contentText = "Hello World!";      // expanded message text

Intent notificationIntent = new Intent(this, MyClass.class);
PendingIntent contentIntent = PendingIntent.getActivity(this, 0, notificationIntent, 0);

// the next two lines initialize the Notification, using the configurations above
Notification notification = new Notification(icon, tickerText, when);
notification.setLatestEventInfo(context, contentTitle, contentText, contentIntent);
```

centic

Notificaciones.

Barra de Notificaciones.

- Notificación sonora

```
notification.defaults |= Notification.DEFAULT_SOUND;
```

```
notification.sound = Uri.parse("file:///sdcard/notification/ringer.mp3");
```

Notificaciones.

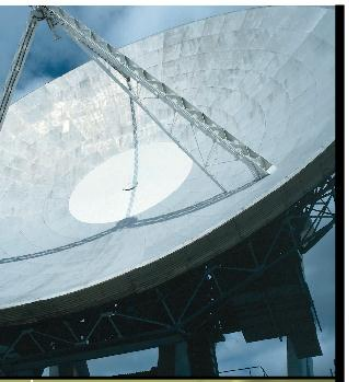
Barra de Notificaciones.

- Notificación Vibrante

```
notification.defaults |= Notification.DEFAULT_VIBRATE;
```

```
long[] vibrate = {0,100,200,300};
notification.vibrate = vibrate;
```

centic



Notificaciones.

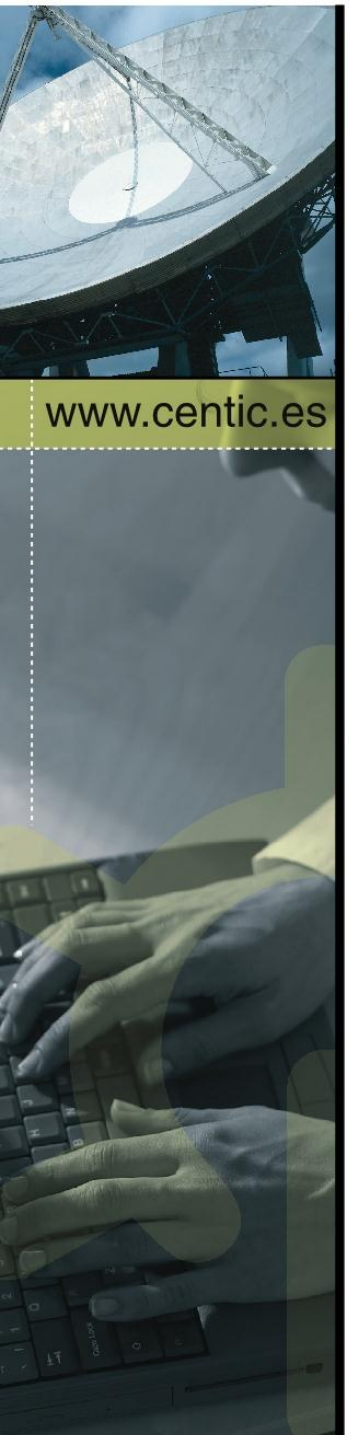
Barra de Notificaciones.

- Notificación Deslumbrante

```
notification.defaults |= Notification.DEFAULT_LIGHTS;
```

```
notification.ledARGB = 0xff00ff00;
notification.ledOnMS = 300;
notification.ledOffMS = 1000;
notification.flags |= Notification.FLAG_SHOW_LIGHTS;
```

centic

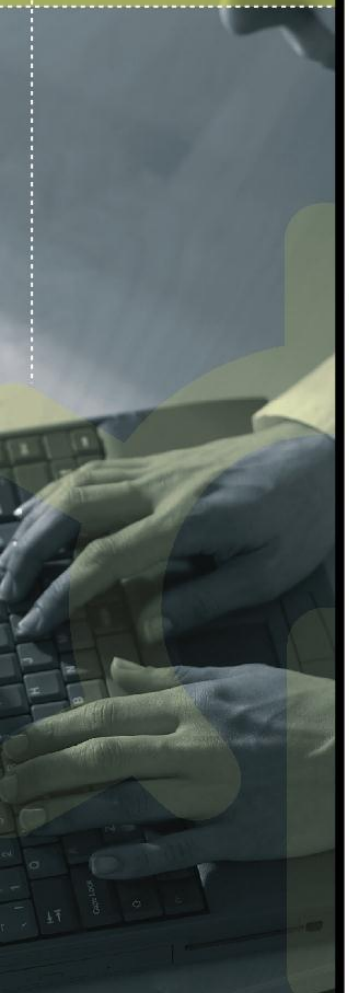


Notificaciones.

Barra de Notificaciones.

- Comportamiento
- Puede agregar varias características más para las notificaciones mediante sus FLAGS
- «FLAG_AUTO_CANCEL»
  - Cancela automáticamente la notificación después de que se selecciona de la ventana de notificaciones
- «FLAG_INSISTENT»
  - Repite el audio hasta que el usuario responde
- «FLAG_ONGOING_EVENT»
  - Indica que la solicitud está en curso
- «FLAG_NO_CLEAR»
  - Indica que la notificación *no* se borrará

centic



Notificaciones.

Barra de Notificaciones.

- Notificación personalizada

  - Creamos su layout

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:orientation="horizontal"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent"
              android:padding="3dp"
              >
    <ImageView android:id="@+id/image"
              android:layout_width="wrap_content"
              android:layout_height="fill_parent"
              android:layout_marginRight="10dp"
              />
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="fill_parent"
              android:textColor="#000"
              />
</LinearLayout>
```

www.centic.es

Notificaciones.

Barra de Notificaciones.

- Notificación personalizada

  - Cargamos el Layout

```
RemoteViews contentView = new RemoteViews(getPackageName(), R.layout.custom_notification_layout);
contentView.setImageViewResource(R.id.image, R.drawable.notification_image);
contentView.setTextViewText(R.id.text, "Hello, this message is in a custom expanded view");
notification.contentView = contentView;
```

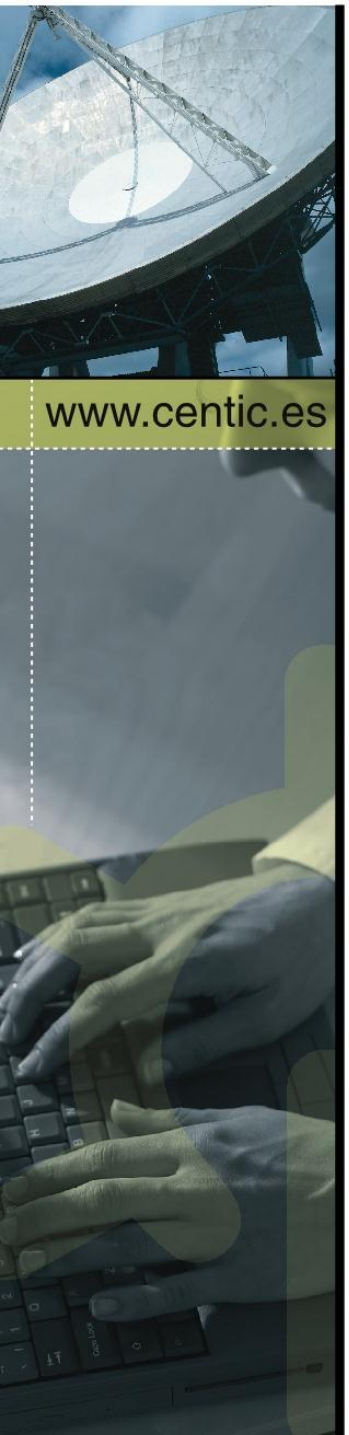
  - Configuramos el «Intent»

```
Intent notificationIntent = new Intent(this, MyClass.class);
PendingIntent contentIntent = PendingIntent.getActivity(this, 0, notificationIntent, 0);
notification.contentIntent = contentIntent;
```

  - Lanzamos la notificación

```
mNotificationManager.notify(CUSTOM_VIEW_ID, notification);
```

centic



Diálogos.

Tipos de Diálogos.

- Dialog
  - Es la clase base para crear cuadros de diálogo
- AlertDialog
  - Un diálogo que puede manejar cero, uno, dos o tres botones
  - Una lista de elementos seleccionables que pueden incluir casillas de verificación o botones de radio.
  - Es capaz de construir la mayoría de interfaces de usuario de diálogo
  - Es el tipo de diálogo recomendado.
- ProgressDialog
  - Un diálogo que muestra una rueda de progreso o barra de progreso
  - También soporta botones

centic

www.centic.es

Diálogos.

Tipos de Diálogos.

- DatePickerDialog
  - Un diálogo que permite al usuario seleccionar una fecha

- TimePickerDialog
  - Un diálogo que permite al usuario seleccionar una hora

centic

www.centic.es

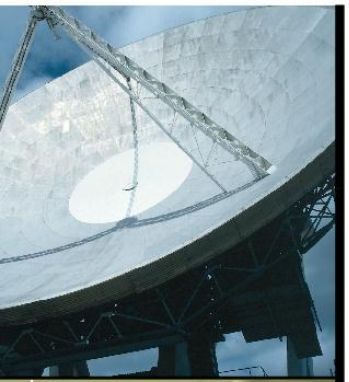
Diálogos.

Tipos de Diálogos.

- DatePickerDialog
  - Un diálogo que permite al usuario seleccionar una fecha

- TimePickerDialog
  - Un diálogo que permite al usuario seleccionar una hora

centic



Diálogos.

Mostrar Diálogos.

- Un cuadro de diálogo siempre se crea y se muestra como parte de una Activity
- Android gestiona automáticamente el estado de cada diálogo
- showDialog(int)
  - Pasarle un entero que identifica únicamente el diálogo que desea mostrar
- onCreateDialog(int)
  - Se llama sólo la primera vez
- onPrepareDialog(int, dialog)
  - Se llama siempre

www.centic.es

Diálogos.

Mostrar Diálogos.

Especificamos los identificadores de cada diálogo

```
static final int DIALOG_PAUSED_ID = 0;
static final int DIALOG_GAMEOVER_ID = 1;
```

centic

Diálogos.

Mostrar Diálogos.

Implementamos el onCreateDialog

```
protected Dialog onCreateDialog(int id) {
    Dialog dialog;
    switch(id) {
    case DIALOG_PAUSED_ID:
        // do the work to define the pause Dialog
        break;
    case DIALOG_GAMEOVER_ID:
        // do the work to define the game over Dialog
        break;
    default:
        dialog = null;
    }
    return dialog;
}
```

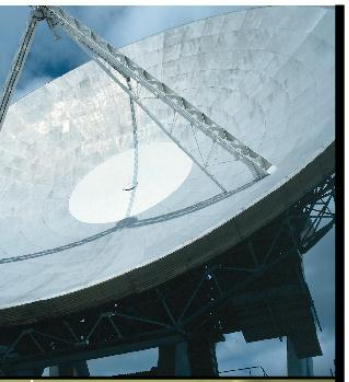
Diálogos.

Mostrar Diálogos.

Implementamos el onPrepareDialog

```
@Override
protected void onPrepareDialog(int id, Dialog dialog) {
    // TODO Auto-generated method stub
     switch(id) {
        case DIALOG_PAUSED_ID:
            // realizar las modificaciones del dialogo
            break;
        case DIALOG_GAMEOVER_ID:
            // realizar las modificaciones del dialogo
            break;
        }
}
```

centic

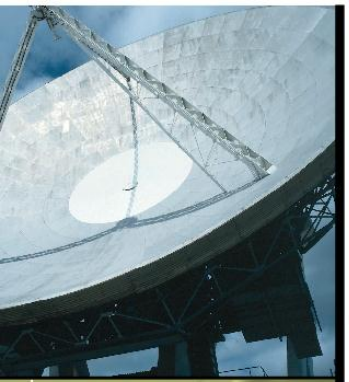


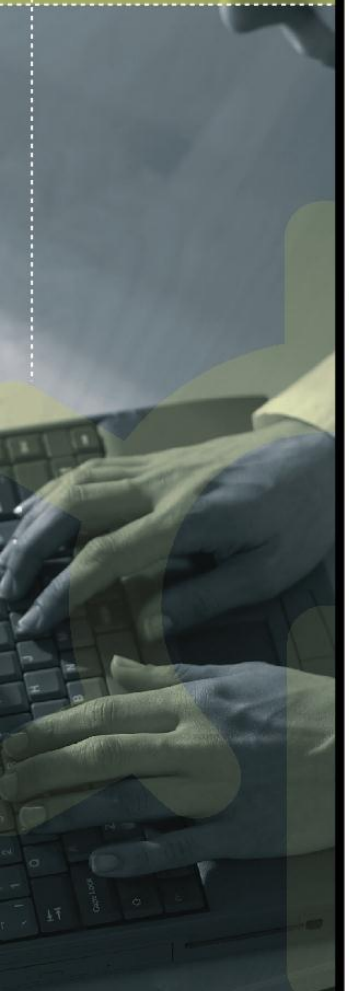
Diálogos.

Mostrar Diálogos.

- Gestionar diálogos
- dismissDialog(int)
  - Oculta el diálogo y mantiene una copia
- removeDialog(int)
  - Elimina el dialogo
- Listener
  - setOnDismissListener()
  - setOnCancelListener()

centic





Diálogos.

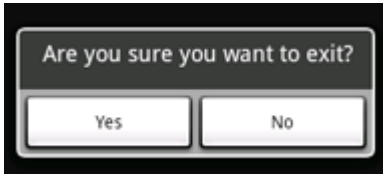
Crear Diálogos.

- Elementos
  - Un titulo
  - Un mensaje de texto
  - Uno, dos o tres botones
  - Una lista de elementos seleccionables
- AlertDialog.Builder
  - create()

Diálogos.

Crear Diálogos.

```
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setMessage("Are you sure you want to exit?")
       .setCancelable(false)
       .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
           public void onClick(DialogInterface dialog, int id) {
                MyActivity.this.finish();
           }
       })
       .setNegativeButton("No", new DialogInterface.OnClickListener() {
           public void onClick(DialogInterface dialog, int id) {
                dialog.cancel();
           }
       });
AlertDialog alert = builder.create();
```



centic

www.centic.es

Diálogos.

Crear Diálogos.

```
final CharSequence[] items = {"Red", "Green", "Blue"};

AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setTitle("Pick a color");
builder.setItems(items, new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int item) {
        //Gestionar elemento
    }
});
```

centic

www.centic.es

Diálogos.

Crear Diálogos.

```
final CharSequence[] items = {"Red", "Green", "Blue"};

AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setTitle("Pick a color");
builder.setSingleChoiceItems(items, -1, new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int item) {
        //Gestionar elementos checkbox
    }
});
AlertDialog alert = builder.create();
```

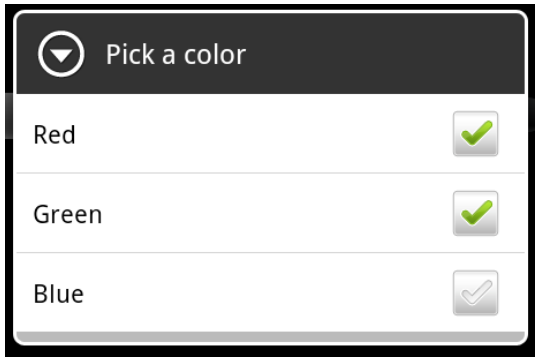
centic

www.centic.es

Diálogos.

Crear Diálogos.

```
final CharSequence[] items = {"Red", "Green", "Blue"};

AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setTitle("Pick a color");
builder.setMultiChoiceItems(items, new boolean[]{true,true,false},

        new OnMultiChoiceClickListener() {

    public void onClick(DialogInterface dialog, int which, boolean isChecked) {
        // TODO Auto-generated method stub

    }
});

AlertDialog alert = builder.create();
alert.show();
```



centic

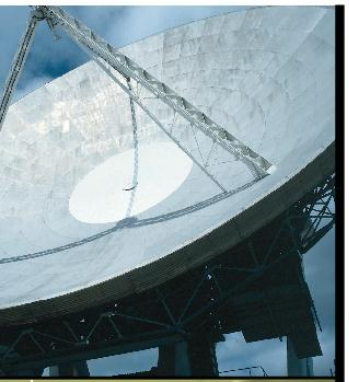
www.centic.es

Diálogos.

Crear Diálogos.

```
ProgressDialog dialog = ProgressDialog.show(MyActivity.this, "",
                        "Loading. Please wait...", true);
```

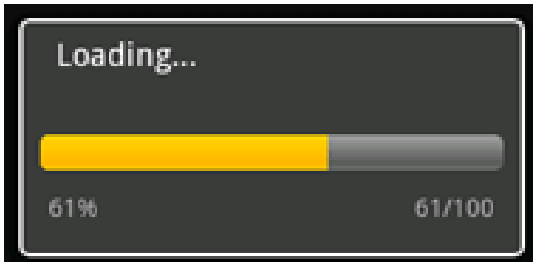
centic



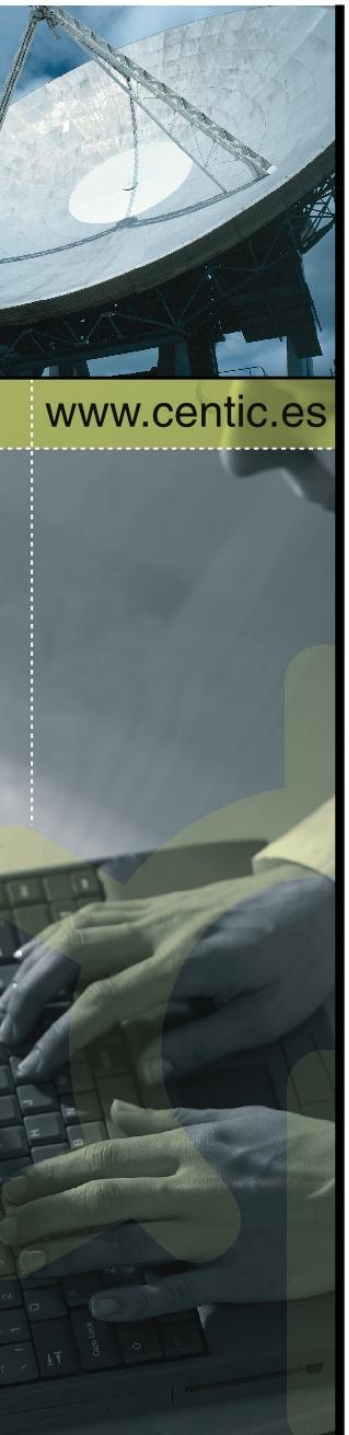
Diálogos.

Crear Diálogos.

```
ProgressDialog progressDialog;
progressDialog = new ProgressDialog(mContext);
progressDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
progressDialog.setMessage("Loading...");
progressDialog.setCancelable(false);
```



centic



Diálogos.

Crear Diálogos.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:id="@+id/layout_root"
              android:orientation="horizontal"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent"
              android:padding="10dp"
              >
    <ImageView android:id="@+id/image"
               android:layout_width="wrap_content"
               android:layout_height="fill_parent"
               android:layout_marginRight="10dp"
               />
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="fill_parent"
              android:textColor="#FFF"
              />
</LinearLayout>
```

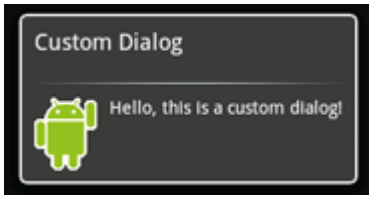
centic

www.centic.es

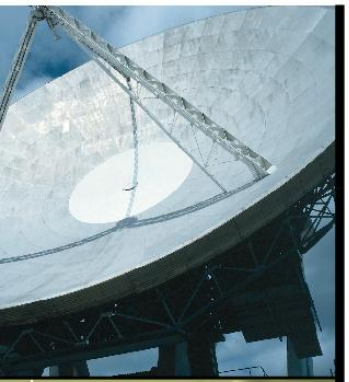
Diálogos.

Crear Diálogos.

```
Context mContext = getApplicationContext();
Dialog dialog = new Dialog(mContext);

dialog.setContentView(R.layout.custom_dialog);
dialog.setTitle("Custom Dialog");

TextView text = (TextView) dialog.findViewById(R.id.text);
text.setText("Hello, this is a custom dialog!");
ImageView image = (ImageView) dialog.findViewById(R.id.image);
image.setImageResource(R.drawable.android);
```



centic

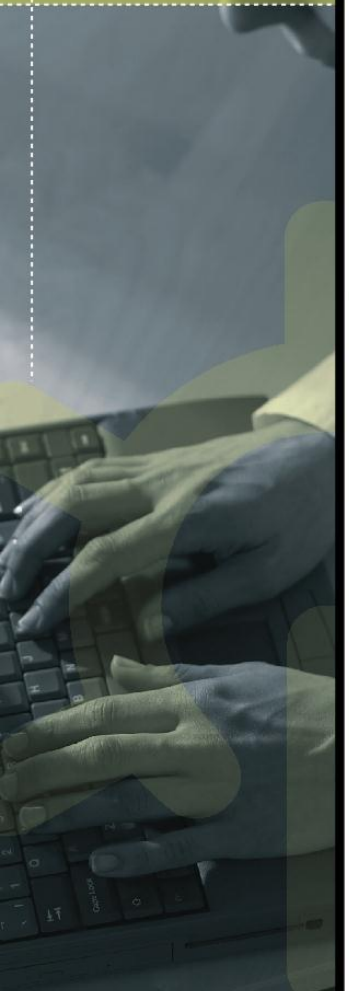


Diálogos.

Crear Diálogos.

```
AlertDialog.Builder builder;
AlertDialog alertDialog;

Context mContext = getApplicationContext();
LayoutInflater inflater = (LayoutInflater) mContext.getSystemService(LAYOUT_INFLATER_SERVICE);
View layout = inflater.inflate(R.layout.custom_dialog,
                               (ViewGroup) findViewById(R.id.layout_root));

TextView text = (TextView) layout.findViewById(R.id.text);
text.setText("Hello, this is a custom dialog!");
ImageView image = (ImageView) layout.findViewById(R.id.image);
image.setImageResource(R.drawable.android);

builder = new AlertDialog.Builder(mContext);
builder.setView(layout);
```

centic



Ejercicios.

Crear Activity para realizar fotos

Crear Activity para reproducir archivos de sonido

Crear menú para lanzar la actividad de fotos, sonidos y salir

Crear dialogo para confirmar que queremos salir de la aplicación

Crear Notificación que avise cada vez que realizamos una foto y nos envíe a la galería de fotos

centic