

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



Empieza el jaleo: publicamos el libro

INFORMÁTICA PROFESIONAL

LAS REGLAS NO ESCRITAS PARA TRIUNFAR EN LA EMPRESA

-¿Crees que la informática es sólo programar?
-¿Sabrías organizar eficientemente un equipo?
-¿Quién te resuelve las dudas sobre esta profesión?..... o te siembra más?

Hosting patrocinado por **enredados**



Inicio **Quienes somos** **Tutoriales** **Formación** **Comparador de salarios** **Comentar libro** **Charlas** **Más**

Estas en: **Inicio** **Tutoriales** Android: Ejemplo usando Widget, BroadcastReceiver y Localización

Ultimas Noticias

- » VII Charla Autentia - Pluto
- » Competición Plasma Cars (Autos Locos) - SEGUNDO INTENTO
- » Probando con Marick - Fotos y vídeo
- » Competición Plasma Cars (Autos Locos) - EVENTO POSPUESTO
- » VI Charla Autentia: Mapeos en Hibernate - Vídeos y Material
- » Competición Plasma Cars (Autos Locos) - EVENTO POSPUESTO

+Noticias Destacadas

- » VII Charla Autentia - Pluto
- » Competición Plasma Cars (Autos Locos) - SEGUNDO INTENTO
- » Probando con Marick - Fotos y vídeo
- » Autentia estuvo en el Spring 2GX Day

+Comentarios Cómic

+Enlaces

Catálogo de servicios Autentia



Triptico (6,3 MB)



Cómic (3,1 MB)

Tutorial desarrollado por



Carlos García Pérez

Creador de [MobileTest](#), un complemento educativo para los profesores y sus alumnos.

Técnico especialista en informática de empresa (CEU)

Ingeniero Técnico en Informática de Sistemas (UPM) *

Puedes encontrarme en [Autentia](#)

Catálogo de servicios de Autentia

Descargar (6,3 MB)

Descargar en versión comic (3,1 MB)

[AdictosAlTrabajo.com](#) es el Web de difusión de conocimiento de [Autentia](#).



[Catálogo de cursos](#)

Descargar este documento en formato PDF: [android_widget_location.pdf](#)

Fecha de creación del tutorial: 2010-04-22

Android: Ejemplo usando Widget, BroadcastReceiver y Localización

Introducción

En este tutorial vamos a realizar un Widget cuya funcionalidad será la de mostrar la calle actual en la que estamos ubicados de manera que cuando hagamos clic en el mismo se nos abrirá un navegador Web con la información que nos provee la wikipedia sobre la persona a la cual se le dedicó la calle.

En el medio de la imagen que se muestra a la derecha se muestra la apariencia visual que tendrá el Widget, de manera que el texto del botón estará en color rojo cuando el GPS esté apagado o fuera de cobertura y en negro en caso contrario cambiándose el texto del botón por el nombre de la calle en la cual nos ubicamos al desplazarnos.

Este ejemplo **servirá para conocer algunos de los conceptos y posibilidades** que nos ofrece **Android**, como por ejemplo:

- Configuración y arquitectura de las aplicaciones en Android
- BroadcastReceiver
- Widget
- Localización
- RemoteViews



Si quieres trastear, puedes descargarte el código fuente [desde clic aquí](#). Si quieres probarlo directamente en tu dispositivo puedes descargarte la aplicación [desde clic aquí](#)

Se presupone que el lector ya tiene ciertos conocimientos sobre Android.

AndroidManifest.xml

A continuación puede observar:

1. Líneas 30 y 31: La aplicación necesita permisos para conectarse a Internet así como tener acceso a un método de localización exacto como el GPS.
2. La aplicación está basada completamente en eventos pues sólo estará viva cuando el sistema lance eventos relacionados con el ciclo de vida de un Widget (APPWIDGET_UPDATE, APPWIDGET_ENABLED, APPWIDGET_DISABLED) o cuando se generen notificaciones relacionadas con el posicionamiento. En este segundo caso, nos hemos creado una acción propia (com.autentia.intent.GPS_LOCATION_CHANGED) y posteriormente le diremos a Android que genere una notificación con ese acción cuando existan eventos de localización. Nos hemos tenido que crear una acción propia, porque no existe ninguna definida en Android para este tipo de notificaciones.
3. Línea 27: La versión mínima de SDK que debe tener el dispositivo es la versión 3 que es la que tienen la gran mayoría (API Level 1.5).
4. Línea 14: Indicamos al WidgetProvider las propiedades del Widget (tiempo de refresco, dimensiones que debe reservar para el mismo, ventana GUI inicial, etc.)

Acceso de usuarios registrados:

E-mail:

Contraseña:

[Deseo registrarme](#)

[He olvidado mis datos de acceso](#)

Registra tu empresa:

Descubre las ventajas de registrar tu empresa en [AdictosAlTrabajo...](#)

[Registrar mi empresa](#)

[Listado de empresas ya registradas](#)



Web

[www.adictosaltrabajo.com](#)

Últimos tutoriales

2010-04-22
[Android: Ejemplo usando Widget, BroadcastReceiver y Localización](#)

2010-04-20
[Facelets en JSF 2: sistema de plantillas y componentes por composición.](#)

2010-04-19
[DbVisualizer free version.](#)

2010-04-09
[Session TimeOut en RichFaces, con el soporte de Jboss Seam.](#)

```

view plain print ?
01. <manifest xmlns:android="http://schemas.android.com/apk/res/android"
02.     package="com.autentia.android.widget.streetinfo"
03.     android:versionCode="1" android:versionName="1.0">
04.
05.     <application android:icon="@drawable/autentia" android:label="@string/applicationName">
06.
07.         <!-- BroadcastReceiver que atender&aacute; las notificaciones relacionadas con el Widget --
08.         <receiver android:name=".StreetInfoWidget" >
09.             <intent-filter>
10.                 <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
11.                 <action android:name="android.appwidget.action.APPWIDGET_ENABLED" />
12.                 <action android:name="android.appwidget.action.APPWIDGET_DISABLED" />
13.             </intent-filter>
14.             <meta-data android:name="android.appwidget.provider" android:resource="@xml/widget_properties"
15.             </receiver>
16.
17.         <!-- BroadcastReceiver para atender las notificaciones de posicionamiento --
18.         <receiver android:name=".LocationUpdatesBroadcastReceiver">
19.             <intent-filter>
20.                 <action android:name="com.autentia.intent.GPS_LOCATION_CHANGED" />
21.             </intent-filter>
22.         </receiver>
23.
24.     </application>
25.
26.     <!-- Versi&oacute;n m&iacute;nima del SDK que debe de tener el dispositivo --
27.     <uses-sdk android:minSdkVersior="3" />
28.
29.     <!-- Permisos que requiere la aplicaci&oacute;n para su funcionamiento --
30.     <uses-permission android:name="android.permission.INTERNET"/>
31.     <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
32.
33. </manifest>
34.

```

res/xml/widget_properties.xml

Después de días enteros acostándome a horas poco normales no he conseguido activar/desactivar el GPS desde programación. Nadie en ningún foro contesta esta pregunta... por favor, si tú sabes como hacerlo, compártelo con todos ;-)
Por cierto, creo que es de las pocas cosas que no puedes controlar desde programación... el resto no he visto limitaciones hasta la fecha.

Bueno, la cuestión es que al colocar el Widget en el escritorio, es posible que el GPS no esté activo, y si no está activo no puedes registrar un oyente de eventos de localización... por lo que cada 20 segundos hago que salte el evento WIDGET_UPDATE y así solvento este problema.

```

view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
03.     android:updatePeriodMillis="20000"
04.     android:initialLayout="@layout/widget_main"
05.     android:minHeight="12sp"
06.     android:minWidth="160sp"
07.     >
08. </appwidget-provider>
09.

```

res/layout/widget_main.xml

La ventana principal del Widget que consta de una imagen con el Logo de [Autentia](#) y un botón en donde se mostrará la calle en la que nos ubicamos.

Si te fijas, en ambos controles gráficos uso estilos (tag style) para configurar la apariencia.

Ojo!!, en los GUI de los **Widgets no puedes colocar cualquier control gráfico** de Android (Spinner, etc) están muy limitados!! (como es lógico pues se supone que es para ver información... que no quita que al hacer clic se abra una ventana de configuración, etc.)

```

view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
03.     android:id="@+id/widget"
04.     android:layout_width="wrap_content"
05.     android:layout_height="wrap_content"
06.     android:orientatior="horizontal">
07.
08.     <ImageView android:src="@drawable/autentia" style="@style/image"/>
09.
10.     <Button android:id="@+id/street"
11.         android:text="@string/currentStreet"
12.         style="@style/command"/>
13. </LinearLayout>
14.

```

res/values/styles.xml

Definición de los estilos... recuerda que esto es un tutorial... tiene miga todo esto, pueden hederarse, etc.

2010-04-08
[Jetspeed-2 de Apache Software Foundation](#)

2010-04-07
[Primeros pasos con Balsamiq Mockups](#)

2010-03-18
[Revisando los ejemplos de Cocos2d para iPhone.](#)

2010-03-16
[Organización de eventos con StageHQ](#)

2010-03-15
[Retrasar la carga de Javascript con jQuery.getScript\(\).](#)

2010-03-15
[Optimización de páginas web con Page Speed.](#)

2010-03-09
[JSF 2 ya está aquí !!! The JSF Return, ahora más sencillo que nunca !!!](#)

2010-03-08
[Instalación de tus programas en tu iPhone.](#)

2010-03-04
[Sacar Release de un proyecto con Maven](#)

2010-03-03
[Instalación de Subversion y Apache en Ubuntu](#)

2010-03-03
[Cómo instalar la JDK de SUN en Fedora Linux](#)

2010-03-02
[Creando un botón de compra de Paypal con datos cifrados](#)

2010-03-01
[Creación de un plugin de tipo hook en Liferay](#)

2010-03-01
[ScrumCards de Autentia en Android](#)

2010-02-25
[Creando la baraja de SCRUM de Autentia como aplicación para Android](#)

2010-02-25
[Instalar CentOS en Virtualbox con NetInstall](#)

2010-02-22
[Expresiones CRON](#)

2010-02-19
[Cómo utilizar el DataStore de Google App Engine con JDO](#)

```

view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <resources>
03.     <style name="command">
04.         <item name="android:textSize">12sp</item>
05.         <item name="android:layout_width">fill_parent</item>
06.         <item name="android:layout_height">wrap_content</item>
07.     </style>
08.     <style name="image">
09.         <item name="android:textSize">12sp</item>
10.         <item name="android:layout_width">wrap_content</item>
11.         <item name="android:layout_height">wrap_content</item>
12.     </style>
13. </resources>

```

res/values/colors.xml

Definición de colores (por supuesto todo esto no es obligatorio...)

```

view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <resources>
03.     <color name="disabled">#FF0000</color>
04.     <color name="enabled">#000000</color>
05. </resources>

```

res/values/strings.xml

En este archivo definimos los recursos de cadenas de caracteres... internacionalización por medio del típico mecanismo de prefijos _es, etc..

```

view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <resources>
03.     <string name="applicationName">Widget StreetInfo</string>
04.     <string name="unknowLocation">Ubicación desconocida</string>
05.     <string name="currentStreet">Ubicación actual</string>
06.     <string name="wikiDomain">http://www.google.com/search?domains=wikipedia.org&hl=es&q=</string>
07. </resources>

```

res/values/arrays.xml

Definición de un array que nos servirá para posteriormente en el código fuente quitar de los resultados de ubicación que nos de el GPS las palabras que contenga este array, por ejemplo si el GPS nos dice que estamos en la calle "Plaza de Manuel Becerra" en el botón mostraremos "Manuel Becerra".

```

view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <resources>
03.     <string-array name="invalidTokensAddress">
04.         <item>Calle de la</item>
05.         <item>Avenida de la</item>
06.         <item>Plaza de la</item>
07.         <item>Calle del</item>
08.         <item>Avenida del</item>
09.         <item>Plaza del</item>
10.         <item>Plaza de</item>
11.         <item>Calle de</item>
12.         <item>Avenida de</item>
13.         <item>Av. de</item>
14.     </string-array>
15. </resources>

```

src/com.autentia.android.widget.streetinfo.StreetInfoWidget.java

El siguiente BroadcastReceiver (Receptor de eventos o notificaciones) será invocado cuando se produzcan los eventos que fueron definidos anteriormente en el archivo de configuración [AndroidManifest.xml](#)

Si el evento es ACTION_APPWIDGET_UPDATE o ACTION_APPWIDGET_ENABLED, solicitaremos el LocationManager que lance la notificación "com.autentia.intent.GPS_LOCATION_CHANGED"... que tratará nuestro siguiente BroadcastReceiver...

En caso de que no sea ninguno de esos dos eventos, será el evento ACTION_APPWIDGET_DISABLED (el usuario ha quitado el Widget de la pantalla u otros motivos) y desregistraremos la notificación de eventos.

Los valores 0,0 de la instrucción requestLocationUpdates... indican el tiempo y los metros transcurridos entre evento y evento.. dejándolos a 0 ahorramos energía, ya que dejamos que el sistema lo gestione y lance eventos cuando crea necesario (si estamos parados no lanzará).

2010-02-19
[Recursos Freeware](#)

2010-02-17
[Plugin de mejora de graficos para JMeter](#)

2010-02-17
[Cómo utilizar el datastore de Google App Engine con su API de nivel inferior](#)

2010-02-16
[Aprendiendo Objective-C desarrollando para nuestro Iphone 3GS](#)

2010-02-11
[Introducción a JCL.](#)

2010-02-09
[Creando la Baraja de SCRUM de Autentia como aplicación para el Iphone 3G.](#)

2010-02-08
[Cómo generar versiones imprimibles de páginas web](#)

2010-02-04
[Como cambiar el tamaño de las fuentes en Xcode \(el entorno de desarrollo para Mac e iPhone\)](#)

2010-02-04
[Primeros pasos con Enterprise Architect y UML 2.x](#)

2010-02-04
[Creación de un componente JSF, basádonos en un plugin de jQuery, con el soporte de RichFaces.](#)

2009-02-03
[Sincronizando el Mail de Mac con Gmail, el correo de Google](#)

2010-02-03
[Integración de jQuery en RichFaces.](#)

2010-02-02
[AjaxSingle: el partialSubmit de RichFaces.](#)

2010-02-01
[Introducción a RichFaces.](#)

2010-01-29
[Transformación de mensajes en SOA con OpenESB](#)

2010-01-26
[JMeter. Uso de funciones.](#)

2010-01-18
[Autenticando los usuarios de Sonar contra un LDAP](#)

2010-01-18
[Introducción a jQuery UI.](#)

```

view plain print ?
01. package com.autentia.android.widget.streetinfo;
02.
03. import android.app.PendingIntent;
04. import android.appwidget.AppWidgetManager;
05. import android.content.Intent;
06. import android.content.Context;
07. import android.location.LocationManager;
08.
09. /**
10.  * Clase principal del Widget
11.  * @author Carlos García. Autentia
12.  */
13. public class StreetInfoWidget extends android.content.BroadcastReceiver {
14.
15.     @Override
16.     public void onReceive(Context context, Intent intent) {
17.         String action = intent.getAction();
18.         PendingIntent locationIntent = PendingIntent.getBroadcast(context,0, new Intent("com.autentia
19.         LocationManager locationManager = (LocationManager) context.getSystemService(Context.LOCATION_S
20.
21.         locationManager.removeUpdates(locationIntent);
22.
23.         if (AppWidgetManager.ACTION_APPWIDGET_ENABLED.equals(action)){
24.             locationManager.requestLocationUpdates (LocationManager.GPS_PROVIDER,0, 0, locationIntent);
25.         } else if (AppWidgetManager.ACTION_APPWIDGET_UPDATE.equals(action)){
26.             locationManager.requestLocationUpdates (LocationManager.GPS_PROVIDER,0, 0, locationIntent);
27.         }
28.     }
29. }

```

src/com.autentia.android.widget.streetinfo.LocationUpdatesBroadcastReceiver.java

Bueno, en esta clase hay más chicha... empecemos

- Esta clase tratará los eventos de localización: GPS habilitado, deshabilitado, temporalmente fuera de servicio y cambio de localización.
- Cada evento tiene unos datos asociados (Bundle).
- Para modificar la vista del Widget y sus controles desde este proceso, usaremos RemoteView. La variable widgetView apuntará a la ventana widget_main y luego con instrucciones como ésta:
`widgetView.setTextColor(R.id.street, context.getResources().getColor(R.color.enabled));` modificaremos uno de los controles que contiene (en este caso modificaremos el color del botón) y posteriormente con la instrucción `manager.updateAppWidget(streetInfoWidget, widgetView)` le diremos al administrador de Widget que lo repinte.
- La clase `android.location.Geocoder` nos servirá para traducir puntos geográficos (latitud, longitud) en direcciones legibles para las personas (calle, país, ciudad, etc).

2010-01-18
[jQuery: cómo crear nuestros propios plugins.](#)

2010-01-18
[Cómo consumir un servicio web RESTful con el soporte de Ajax y JSON de jQuery.](#)

2010-01-18
[Introducción a jQuery.](#)

2010-01-17
[Introducción a Tapestry 5](#)

2010-01-14
[JMeter. Gestión de usuarios](#)

2010-01-14
[Patrón Visitor con commons-collections y sus Closures](#)

2010-01-12
[Creación de servicios web Restful, con soporte a persistencia, en NetBeans.](#)

2010-01-11
[JMeter y JSF. Extracción del parámetro ViewState](#)

2010-01-07
[Importar el correo de Microsoft Outlook al cliente de correo de Mac OS.](#)

2010-01-07
[Monitor de Hudson para Eclipse.](#)

Últimas ofertas de empleo

2009-07-31
[T. Información - Operador \(día / noche\) - BARCELONA.](#)

2009-06-25
[Atención a cliente - Call Center - BARCELONA.](#)

2009-06-19
[Otras - Ingenieria \(minas, puentes y puertos\) - VALENCIA.](#)

2009-06-17
[Comercial - Ventas - ALICANTE.](#)

2009-06-03
[Comercial - Ventas - VIZCAYA.](#)

Ads by Google

```

view plain print ?
01. package com.autentia.android.widget.streetinfo;
02.
03. import java.util.List;
04. import android.app.PendingIntent;
05. import android.appwidget.AppWidgetManager;
06. import android.content.ComponentName;
07. import android.content.Intent;
08. import android.content.Context;
09. import android.location.Address;
10. import android.location.Geocoder;
11. import android.location.Location;
12. import android.location.LocationManager;
13. import android.location.LocationProvider;
14. import android.net.Uri;
15. import android.os.Bundle;
16. import android.widget.RemoteViews;
17.
18. /**
19.  * Recibirá notificaciones de Localización
20.  * @author Carlos García. Autentia
21.  */
22. public class LocationUpdatesBroadcastReceiver extends android.content.BroadcastReceiver {
23.
24.     @Override
25.     public void onReceive(Context context, Intent intent) {
26.         Bundle data = intent.getExtras();
27.         RemoteViews widgetView = new RemoteViews(context.getPackageName(), R.layout.widget_main);
28.         boolean isGPSProviderEnabled = true;
29.
30.         try {
31.             // Analizamos los valores que nos entregan (dependerá del evento)
32.             if (data.containsKey(LocationManager.KEY_LOCATION_CHANGED)){
33.                 Location point = (Location) data.get(LocationManager.KEY_LOCATION_CHANGED);
34.                 Geocoder geocoder = new Geocoder(context);
35.                 List<Address> addrs = geocoder.getFromLocation(point.getLatitude(), point.getLongitude(), 1);
36.                 String street = this.parseAddress(context, addrs.get(0).getAddressLine(0)); //
37.
38.                 // Modificamos el TextView con el contenido de la calle
39.                 widgetView.setTextViewText(R.id.street, street);
40.
41.                 // Modificamos el comportamiento del Button para que cuando hagamos clic en
42.                 // se abra un Activity con capacidad de mostrar el contenido de una UR
43.                 String domain = context.getResources().getString(R.string.wikiDomain);
44.                 PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, new Intent(Intent.ACTION_VIEW, Uri.parse(domain + street)));
45.                 widgetView.setOnClickPendingIntent(R.id.street, pendingIntent);
46.
47.             } else if (data.containsKey(LocationManager.KEY_STATUS_CHANGED)){
48.                 isGPSProviderEnabled = (data.getInt(LocationManager.KEY_STATUS_CHANGED) == LocationManager.PROVIDER_LOCATION_SERVICE);
49.             } else if (data.containsKey(LocationManager.KEY_PROVIDER_ENABLED)){
50.                 isGPSProviderEnabled = data.getBoolean(LocationManager.KEY_PROVIDER_ENABLED);
51.             }
52.         } catch (java.io.IOException e) {
53.             // No se dará
54.         }
55.
56.         // Cuando el GPS esté en On y haya cobertura el botón tendrá un estilo u otro
57.         if (isGPSProviderEnabled){
58.             widgetView.setTextColor(R.id.street, context.getResources().getColor(R.color.enabled));
59.         } else {
60.             widgetView.setTextColor(R.id.street, context.getResources().getColor(R.color.disabled));
61.         }
62.
63.         // Actualizamos la ventana del widget
64.         ComponentName streetInfoWidget = new ComponentName(context, StreetInfoWidget.class);
65.         AppWidgetManager manager = AppWidgetManager.getInstance(context);
66.
67.         manager.updateAppWidget(streetInfoWidget, widgetView);
68.     }
69.
70.
71.
72.     /**
73.     * @return Dada una dirección con el formato "Calle de Pedro de Alarcón, 30" devuelve
74.     */
75.     private String parseAddress(Context context, String addressLine) {
76.         int pos = addressLine.indexOf(",");
77.
78.         if (pos != -1){
79.             addressLine = addressLine.substring(0, pos);
80.         }
81.
82.         String[] addrTokens = context.getResources().getStringArray(R.array.invalidTokensAddress);
83.         for (int i = 0, count = addrTokens.length; i < count; i++){
84.             addressLine = addressLine.replaceAll(addrTokens[i], "");
85.         }
86.         return addressLine;
87.     }
88. }

```

Conclusiones

Sin lugar a dudas, **Android es sorprendente...** nos ofrece un abanico de posibilidades inmensas, está bien diseñado y es relativamente fácil de programar..

En **Autentia**, estamos constantemente formándonos para conocer cuáles son las herramientas que hay en el mercado y cuáles se adaptan mejor en cada caso. Espero nos tengais en cuenta si necesitais algún tipo de consultaría o formación a medida.

¿Qué te ha parecido el tutorial? Déjanos saber tu opinión y ivota!

Muy malo Malo Regular Bueno Muy bueno



Votar

(Sólo para usuarios registrados)

» [Regístrate y accede a esta y otras ventajas](#) «

Anímate y coméntanos lo que pienses sobre este tutorial

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» [Regístrate y accede a esta y otras ventajas](#) «

Autor **Mensaje de usuario registrado**

- Puedes inscribirte en nuestro servicio de notificaciones haciendo clic [aquí](#).
- Puedes firmar en nuestro libro de visitas haciendo clic [aquí](#).
- Puedes asociarte al grupo AdictosAlTrabajo en XING haciendo clic [aquí](#).
- Añadir a favoritos Technorati.  ADD THIS BLOG TO MY **Technorati FAVORITES**



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos

...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com

Formación en nuevas tecnologías

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf
Android: Ejemplo usando Widget, BroadcastReceiver y Localización	En este tutorial vamos a realizar un Widget cuya funcionalidad será la de mostrar la calle actual en la que estamos ubicados de manera que cuando hagamos clic en el mismo se nos abrirá un navegador Web con información que nos provee la wikipedia...	2010-04-22	9	-	-	
Primeros pasos con Balsamiq Mockups	En este tutorial veremos los primeros pasos de una herramienta estupenda para hacer borradores rápidos de Webs o incluso aplicaciones móviles: Balsamiq Mockups	2010-04-07	453	-	-	
Revisando los ejemplos de Cocos2d para iPhone.	En tutoriales anteriores hemos descubierto cómo empezar a trabajar con el iPhone. Para construir un juego supongo que es una locura plantearse hacerlo desde cero cuando hay tanta gente que generosamente nos ofrece su talento..	2010-03-18	850	-	-	
ScrumCards de Autentia en Android	En este tutorial usaremos los dispositivos móviles de un modo eficiente. Para poder ser justos en la comparación de las dos principales tecnologías: iPhone y Android vamos a desarrollar la misma aplicación a la desarrollada sobre la baraja de Scrum	2010-03-01	1113	-	-	
Creando la baraja de SCRUM de Autentia como aplicación para Android	En este tutorial veremos cómo montar el entorno para desarrollar con Android y cómo hacer una aplicación un poco más completa que un simple "Hola mundo"	2010-02-25	1376	Bueno	1	
Aprendiendo Objective-C desarrollando para nuestro Iphone 3Gs	En este tutorial veremos que aunque el lenguaje y entorno para el Iphone puedan sernos totalmente nuevos hay decenas de posibles combinaciones con las aplicaciones empresariales que habitualmente nos piden.	2010-02-16	1512	-	-	
Creando la Baraja de SCRUM de Autentia como aplicación para el iPhone 3G.	En este tutorial, se me ha ocurrido que podría hacer una pequeña aplicación útil: el pasar a iPhone la baraja de estimación que utilizamos en nuestra reuniones Scrum	2010-02-09	1154	-	-	
JSR-179 Location API para J2ME: Posicionamiento geográfico en nuestras aplicaciones.	En este tutorial Carlos García nos presentará la especificación JSR-179 Location API para J2ME, en donde veremos como realizar aplicaciones con funcionalidad relacionada con el posicionamiento geográfico.	2009-07-19	6234	Muy bueno	3	
LWUIT: Una librería gráfica tipo AWT o Swing para J2ME	En este tutorial Carlos García os presenta una librería para J2ME con mucha similitud con Swing o AWT.	2009-06-10	6513	-	-	
JTAPI. El API de Telefonía para Java	En este tutorial veremos varios ejemplos del uso de JTAPI.	2009-04-13	5162	-	-	

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.