<u>Cookbook</u> Creando un Proyecto Android (ADT-Eclipse)

ALONSO PARRA CESAR VIELMA FREDDY RONDON JOSE MARQUEZ Alienx9889 * cesarvielma * spantons * joseangel2212

*

* @gmail.com

Universidad de Los Andes Escuela de Ingeniería de Sistemas La Hechicera, Mérida 5101

RESUMEN: En el siguiente artículo explicaremos como crear un nuevo proyecto Android con las herramientas ADT y Eclipse. **PALABRAS CLAVES:** Android, Proyecto, Hola, Mundo.

ABSTRACT: In the next article we will explore how to create a new Android project with the ADT and Eclipse tools.

Keywords: Android, Project, Hello, World.

I. INTRODUCCION

Primero y principal para realizar un Proyecto Android necesitamos de una serie de herramientas de suma importancia para que nuestra nueva aplicación se pueda ejecutar y posteriormente llevarla a nuestro teléfono inteligente.

Esas herramientas son (Para este tutorial): SDK Android y Eclipse IDE.

El Primero, SDK Android es un software que nos proporciona las APIs (Bibliotecas) y herramientas para el desarrollo, ejemplo: Creación, Prueba y Depuración de aplicaciones para Android. Para comenzar como desarrollador Android es indispensable descargar el paquete de ADT, con esto incias rápidamente el desarrollo de aplicaciones.

Este SDK incluye cosas importantes para comenzar a desarrollar aplicaciones:

1. CREANDO EL PROYECTO

PASO 1.- AJUSTES/REQUERIMIENTOS

Eclipse + ADT Plugin Android SDK Herramientas Android Platform-tools

Como se observa el SDK incluye un IDE de Eclipse, con esto no seria necesario descargar otro IDE Eclipse.

Para descargar el SDK Android dirígete a la siguiente dirección:

http://developer.android.com/sdk/index.html es multiplataforma: Windows, Mac OS y Linux. El primer paso para crear un nuevo proyecto Android, suponiendo que tenemos todas las herramientas básicas instaladas y corriendo a la perfección, haciendo clic derecho en el Package Explorer del ADT se nos desplazará un menú (ver Figura 1). Allí seleccionaremos la opción New y luego nos mostrará los tipos de proyectos que podemos realizar, en nuestro caso será Android Application Project.



Figura 1.

PASO 2.- PERSONALIZANDO NUESTRO PROYECTO

En la ventana que se nos presenta a continuación (Ver Figura 2), debemos indicar el nombre de la Aplicación, Nombre de nuestro Proyecto y Nombre del Paquete (Esto es la forma en que Android identifica el Proceso).

Posterior a eso nos encontramos con: Minimum Required SDK (SDK mínimo requerido), es la versión mínima de Android que soporta tu aplicación, especificada usando los Niveles API. Para que tu aplicación funcione en todos los dispositivos Android, deberías establecer el valor al API mínimo.

Target SDK (SDK objetivo) es la versión más moderna de Android con la que puedes probar tu aplicación, esto sirve para verificar si tu aplicación es compatible con Apis mas actualizadas o recientes.

Compile With (Compilar con) es la versión de la plataforma contra la que quieres compilar tu aplicación.

Theme (Tema) especifica el estilo de interfaz de usuario Android que aplicar a tu aplicación. No es necesario modificar esta opción.

| Index Deformer Control of Control o |
|---|
| It includes polymer XI It includes p |
| Implementage Implementage< |
| B adorective cores 6 B Core Core B Core Fill B Core Project Neme ¹ (Field Mode) B Core Project Neme ¹ (Field Mode) |
| By Current Application Name: Point |
| Constantiation Applications |
| Project Name: 0 HoleMundo |
| |
| R 🗟 Gooldtification Package Name: 1 con.example.holamundo |
| B 20 Gooderver |
| Minimum Required SDI(3) PEB. Android 2.2 (Eroyo) |
| # 20 Instituctionmanager Target SDK: 0 API 17: Android 4.2 (Jelly Bean) |
| B B Pholog Comple With: 0 AP1 17: Android 4.2 (Xely Bean) |
| ar ⊇r puebas ⊇ Carenda pueba table Theman () while Links with Dueb Artisian Bas |
| a le neurospic de la constancia de la c |
| B 😹 Servidor |
| Ghose the lowest version of Andreits that your application will support. Lower API levels target more devices, but meet the metal in the support of the metal in the support. Lower API levels target more devices, but metal in the metal in the support. |
| 8 (p) test |
| |
| B 😂 TestustExpandable |
| B 2 TestPRobg |
| |
| |
| B 2 TutorialAlamManager DDMS Cancel |
| |
| |
| |
| |
| |
| |
| - 2 |
| |
| 105% of 2894 |
| |
| |

Figura 2.

PASO 3.- CONFIGURANDO ICONO PARA EL LAUNCHER

En esta parte podemos personalizar el icono para nuestra aplicación ó como en este caso (*Ver Figura 3*) dejar el que está por defecto.

| 🚯 Java - ADT | | X |
|--|--|------------------|
| File Edit Refactor Source Navigate Search Project Run V | indow Help | (|
| 📑 • 🖾 🖄 📥 🚺 🖬 🗹 • | Vew Android Application | Quick Access 🗈 😰 |
| It Package Gutere 21 C C C C C C C C C C C C C C C C C C | Configure lander for Configure lander for Configure lander for Integrand: Support The support is lander for The support lander is land Alticular lander is Support lander is Sup | |
| | | |

Figura 3.

PASO 4.- CREAR LA ACTIVITY PRINCIPAL

Para poder agregar un interfaz de usuario a nuestra aplicación necesitamos una clase "Activity", ya que esta es la base para cualquier aplicación Android con interfaz.

Una aplicación puede tener más de una pantalla visible, es por eso que cada activity está relacionada con una pantalla visible.

En nuestro ejemplo inicial seleccionaremos una BlackActivity, sencilla para nuestra aplicación (Ver Figura 4).

| 🚯 Jaya - ADT | | | | | | | _ 5 > |
|---|----------|--|--------------------------------|---|------------|-------------------------------|----------|
| File Edit Refactor Source Navigate Search Project R | un Windo | Help | | | | | |
| 🖬 • 🗉 🗛 📥 👘 🗖 🗖 | - 🚮 | New Android Application | | | _IO × | Quick Access | evet 🐉 😭 |
| 12 Backage Evoluter 22 | | Create Activity | | | | PE outine 32 | |
| | | Select whether to create an activity, and | if so, what kind of activity. | | | An outlos is not available | |
| E 🥵 andraid actificationar | | | | | | ALL COLUMN IN THE OTHER DATE: | |
| R Gerte | | Create Artivity | | | | | |
| E Departamento | | North Antonio | | | | | |
| E 🦉 Goo | | Fulscreen Activity | | _ | | | 4 |
| GooAsistenteClente | | Master/Detail Flow | | | | | - |
| 🕀 😂 GooNotification | | | | | | | - |
| E 😥 GooServer | | | | (| | | ~ |
| B Image Inotificaciones | | | | | | | |
| E 🔛 Notification | | | | | | | |
| 🗄 🥁 notificationmanager | | | | | | | |
| E Prolog | | | | | | | |
| 🕒 🔛 pruebas | | | | | | | |
| er er Fruebkaspek | | | | | | | |
| is poster | | | | | | | |
| E Servicor | | | | | | | |
| R-Se test | | | | | | | |
| F Sa testd | | | | | | | |
| E 🛃 TestGoo | | | | | | | |
| 🕑 🥁 TestListExpandable | | Direct Anthony | | | | | |
| 🗄 🥵 TestPRolog | | Blank Activity | | | | | |
| 🖻 🍰 TextList | | Creates a new blank activity, with an acti | on bar and optional navigation | al elements such as tabs or horizontal swipe. | · | | |
| 🖲 🍰 TL | Conv. | | | | - · · · · | | |
| 🕀 🍪 TLE | CO DC | | | | | | |
| TutorialAlarmManager | 00%6 | | | | | 3 | |
| | | 0 | r Paul | Mart > Dick | - | | |
| | | | | INEXT AND C | aite | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | 2 | 1 | |
| | ₹ | | | | <u>}</u> | | |
| | | | | | 170M of 25 | SM 🣋 | |
| | | | | | | | |
| | | | 2 | | | | |
| | | | 3 | | | | |
| | | | 1 | | | | |
| | | | | | | | |

Figura 4.

PASO 5.- PERSONALIZAR ACTIVITY PRINCIPAL

Aquí simplemente personalizamos los datos de la activity, el nombre de su Layout y el tipo de navegación.

| 🕦 Java - ADT | | | | × |
|---|------------|--|--------------|------------------------------|
| File Edit Refactor Source Navigate Search Project R | un Winde | w/ Help | | |
| | a 🗸 🙀 | 🚯 New Android Application 📃 🗆 🗙 | | Duick Access |
| | - 164 [| Blank Activity | | |
| A Package Explorer 🛛 📄 🦉 👋 🗖 🗖 | | Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal | | E Outline X |
| 🕀 🙀 Alarmanager | | swipe. | | An outline is not available. |
| 🗄 🎬 android-notificaciones | | | | @ |
| B B Cliente | | | | |
| B Car | | | | 1 |
| B Goot distante Cleante | | | | |
| E GooNotification | | Activity Name® MainActivity | | |
| E 😂 GooServer | | Layout Name® activity main | | |
| 🗄 🙀 notificaciones | | No desider Tors 0 2000 | | |
| 🕀 🥰 Notification | | Nengadon Type V KKE | | |
| 🗄 🚰 notificationmanager | | | | |
| E prolog | | | | |
| E pruebas | | | | |
| H Server | | | | |
| 🗄 😒 Servidor | | The type of navigation to use for the activity | | |
| 🗄 🙀 TaskManager | | | | |
| 🕀 💕 test | | | | |
| 🗄 😥 testd | | | | |
| 🗄 🙀 TestGoo | | | | |
| 🗄 🐷 TestListExpandable | | | | |
| B S Testing | | | | |
| H-SS TI | | | | |
| 🕀 🍰 TLE | Conso | | | |
| 🗄 🤪 TutorialAlarmManager | DDM5 | | | |
| | | | <u> </u> | |
| | | < gack (yew > Ensin Cancel | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | Figura 5 | | |
| | ₹ | | | |
| | _ | | 133M of 271M | la la |

PASO 6.- NUEVO PROYECTO CREADO

Al pulsar el botón finalizar en el paso anterior, tendremos listo nuestro nuevo proyecto. Podemos observar que se genero la activity y el Layout con los nombres que usamos, ¡Ya tenemos nuestro primer Hola Mundo!. Pero ahora ¿Como lo ejecutamos en nuestro emulador ó teléfono?

| 🚯 Java - HolaMundo/res/layout/activity_main.xml | DT | | | _ 8 |
|---|---|------|---|---------------|
| File Edit Refactor Source Navigate Search Project | | | Decision of the second s | |
| | | | I CUICK ACCESS | EV I Stave |
| 🖬 Package Explorer 33 📄 🧐 🗖 | C activity_main.xmi 23 | | Bt Outline 33 | |
| B-2 HoleMundo | Palette 2 ▼ □ Nexus One ▼ 2 ▼ ☆ AppTheme ▼ ⊙ MainActivity ▼ § ▼ ↓ | 17 • | | orid1" |
| com.example.holamundo | 🕞 Form Widgets \cdots 🗓 😳 🔹 🐌 💭 🔍 | | | 8 |
| ManActivity.java | Textilev Large Medium snat | | | |
| Gran gen (Generated Java Files) | | _ | | |
| E BuildConfig.tava | | | | |
| E R.java | Checkbox Radiobutton | | | |
| E Android 4.2.2 | Checked TextTrikew | | | |
| Android Dependencies | Spinner Helio world! | | | |
| - assets | | | | |
| te 😂 bin | | | | |
| android-support-ust iar | | | | |
| R-Sa red | | | | |
| 🖲 😂 drawable-hdbi | | | | |
| - 🗁 drawabile-idpi | | | | |
| 🖲 🎃 drawable-mdpi | | | | |
| 🗈 🎃 drawable-xhdpi | | | 📼 Properties 🏾 🎲 🧏 | E 🗉 🗉 |
| 🕀 🗁 drawable-xohdpi | | | Id | |
| k_launcher.png | | | * Layout Par [] | |
| E G Ayour | | | Background Badding Laft Odmoving | |
| acovey_man.xm | C Text Fields | | Content De | WEY_DEFL |
| | Carouts | | RelativeLayout [] | _ |
| e values | | | Gravity | |
| 🕀 🗁 values-sw600dp | | | - Ignore Gr | |
| e walues-sw720dp-land | i images & Media | | Sy/e | - |
| 🖲 🎃 values-v11 | 🖸 Time & Date | | Tag | = |
| E 🗁 values-v14 | C Transitions | | Background | |
| - C AndroidPlanifest.xm | C Advanced | | Paging Left (Indimen/acti | inter bori un |
| - Mic_launcher-web.png | C Other | | Padding Top @dmen/acti | Mty_vert |
| noiert moneties | Custom & Ubrary Yiews | | Padding R @dmen/acti | Mty_hori |
| · · · · · · · · · · · · · · · · · · · | | | Padding B @dimen/acti | wky_vert |
| | Arabura rayon: F. acouch Tuaurous | | rocusabié | <u> </u> |

Figura 6.

3. EJECUTANDO EL PROYECTO

Existen varias formas para ejecutar un proyecto Android, la primera seria por el emulador del AVD (Android Virtual Device) y la segunda usando nuestro dispositivo en modo Depuración USB. Explicaremos brevemente cada uno.

3.1 VIA ANDROID VIRTUAL DEVICE

Nos dirigimos y pulsamos AVDM (Ver Figura),

| \odot | Java - | HolaMun | do/res/l | ayout/act | ivity_m | ain.xml - | ADT | | | | | | |
|---------|--------|-------------|----------|-----------|---------|-----------|----------|--------------|----------|-----|------------|-----|-----|
| File | Edit | Refactor | Source | Navigate | Search | Project | Run | Window | Help | | | | |
| 2 | • 🖁 | 16 b | | | | ē 😑 | • | d 🛛 | 称• | • • | Q _ | - # | 8 (|
| 1 | Packag | e Explorer | x | | 🐴 🔉 | | idroid \ | /irtual Devi | ce Manaç | per | | | |
| ė | 😂 He | olaMundo | | | | - | • | ==== Pa | lette = | = | | a | • |

se nos mostrara la siguiente ventana (Ver Figura).



Pulsaremos nuevo para crear un nuevo Virtual Device. Aquí personalizaremos nuestro emulador, en nuestro caso elegiremos en Target Android 2.2, debemos tener en cuenta que si en nuestra creación de proyecto elegimos *Minimum Required SDK* API 10, no podemos elegir el API 8 por ejemplo, ya que no será compatible y no reconocerá el emulador.:

| Android Virtual Dev | vices Device Definitions | 🕕 Create new An | ndroid Virtual Devic | e (AVD) | 1 |
|--|---|----------------------------------|---|--|------------------------------------|
| List of existing Ar | droid Virtual Devices loca | AVD Name: | Emulador | | |
| AVD Name | Target Name No AVD available | Device: | 4.65" 720p (720 × | 1280: ×hdpi) | |
| | | Target: | Android 2.2 - API Le | vel 8 | |
| | | CPU/ABI: | ARM (armeabi) | | v |
| | | Keyboard: | Hardware keybo | and present | |
| | | Skin: | Display a skin wit | h hardware control | ß |
| | | Front Camera: | None | | - |
| | | Back Camera: | None | | - |
| | | Memory Options: | RAM: 1024 | VM Heap: 64 | |
| | | Internal Storage: | 200 | | мв 💌 |
| | | SD Card: | G Size: | | MB - |
| A valid Andro X An Android V | id Virtual Device. 🔂 A irtual Device that failed t | | C File: | | Browse |
| values-sweett values-sweett values-sw720 | op dp-land | Emulation Options: | Snapshot | Use Host GPU | |
| e 🕗 values-v11 | | I TOYETTAE DIE EX | - and the second second | | |
| AndroidManifest.: | cml | On Windows, e Try progression | emulating RAM greate elv smaller values of F | r than 768M may fa AM if the emulator | al depending of Fails to launch |

Una vez creado el emulador procedemos a ejecutar nuestro proyecto:

| 🚯 Java - Ho | olaMundo/res/layout/activit | y_main.xml - | ADT |
|-------------|--|---------------|---|
| File Edit R | efactor Source Navigate Se | earch Project | Run Window Help |
| 📑 👻 🖂 🛛 | | A 🖻 | 🛛 • 🗟 🔌 🕸 • O • O. • 🖶 🖓 • 🖄 |
| | | | |
| Package E | Explorer 🛛 📃 🤤 | , · · · | activity_main.xml 🛛 |
| 🖻 😂 Hela' | New | | Palette a v Nes |
| E 🖉 : | Go Into | | ② Palette |
| | Onen in New Window | | 🗁 Form Widgets 🔛 🔛 🛄 |
| E 😕 (| Open Type Hierarchy | F4 | TextView Large Medium Small |
| - B-1 | Show In | Alt+Shift+W | Button Small OFF |
| | E Conu | CHLC | CheckBox BadioButton |
| | Copy Cushing Name | curre | CheckedTextView |
| | Deste | CHUN | Spinner Hello wo |
| 1 5 | Paste State | Cori+v | Sub Item |
| i ⊕ 🍒 i | × Delete | Delete | |
| 0-🍰 I | Build Path | 1 | |
| | Source | Alt+Shift+S | |
| | Refactor | Alt+Shift+T | |
| | 🚵 Import | | |
| ÷- | 🛃 Export | | |
| | A Pefrech | 65 | |
| B-(| Close Project | 10 | <u>A A A A A</u> |
| | Close Unrelated Projects | | 000 |
| | Assign Working Sets | | |
| | Run As | | 2 1 Android Application |
| | Debug As | | 2 Android 1 Init Test |
| | Profile As | | T 3 Java Anglet Alt+Shift+X. A |
| ÷. | Validate | | 4 Java Application At+Shift+X 1 |
| E-1 | Team | | Ty 5 1102 Text Alt+Shift+V T |
| 100 H | Compare with Restore from Local History | | |
| | Android Tools | | Run Configurations |
| - 🔂 i | ProDT | | Advanced |
| 🗎 p | Properties | Alt_Enter | 🛄 Other |
| 1 | Resource Configurations | HICHLING | Eustom & Library Views |
| | | | 🔚 Graphical Layout) 🔚 activity_main.xml |
| HolaMundo | | | |

Al ser ejecutado el proyecto se verá en el emulador algo como esto:

| 🔛 🚮 📧 7:17 PM | | | | | | | | | | |
|---------------|-----|----------------|----|------|------------|-----|------------|----|-----|-----|
| Hola Mundo | | | 6 | | | 0 | | Ċ | | |
| | | | 6 | | it's | R | | • | | |
| | | | | | AENU | 6 | | 9 | | |
| _ | 1 | 2 ⁰ | 2# | A \$ | 5 % | 6 ^ | 7& | 0 | 0 (| 0) |
| | Q | W | E | R | т (| Y) | <i>U</i> - | I | 0 | P |
| | A | s | D | F | G | Н < | 1 > | К | L | DEL |
| | 슣 | Z | X | С | V | В | N | M | | ł |
| | ALT | SYM | @ | | - | _ | →I | 1? | | ALT |

3.2 VIA DEPURACION USB

Para ejecutar un proyecto mediante modo Depuración USB tenemos que configurar nuestro dispositivo de la siguiente manera (Esto varía dependiendo de la versión Android del dispositivo).



Una vez conectado el dispositivo vía USB y configurado como se mostro en la imagen anterior, procedemos a ejecutar nuestro proyecto:

| 🚯 Java - He | olaMundo/res/layout/activi | y_main.xml - i | ADT | |
|---------------------------------------|--|----------------|--------------------------------------|------------|
| File Edit R | efactor Source Navigate Si | earch Project | Run Window Help | |
| 📑 🕶 🔡 🛛 | | 🛅 🗎 I | 🗹 • 🔮 🔌 🏇 • 🔕 • 🂁 • 🔀 🞯 • | 2 |
| 📮 Package I | Explorer 🛛 📄 😫 | ~ - 8 | activity_main.xml 🛛 | |
| 🖻 🥵 Hola | | 100 | 🕴 💳 Palette 💳 👘 💿 | |
| ÷ 🥮 : | New Co Inho | , | ③ Palette 🗢 🔍 🍟 | Ne: |
| B- | Gonico | | 🗁 Form Widgets 🕀 🗊 | |
| - Sa | Open in New Window | 54 | TextView Large Medium Small | - |
| | Show In | Alt+Shift+W | Button Small OFF | |
| | 500121 | | | ا ر |
| | Copy | Ctrl+C | CheckBox RadioButton | |
| E 🛋 / | Copy Qualified Name | | Soippor | |
| 1 🔁 🐴 / | Paste | Ctrl+V | Sub Item | wo |
| | 💢 Delete | Delete | | |
| i ⊡≦i | Build Path | • | | |
| | Source | Alt+Shift+S 🕨 | | |
| 🖻 🦢 ı | Refactor | Alt+Shift+T 🕨 | | |
| | 🚵 Import | | | |
| | 🛃 Export | | | |
| Ē. | A Defeet | | | |
| B-(| Close Project | 15 | A A A A A | |
| | Close Unrelated Projects | | OFF | |
| B-(| Assign Working Sets | | | |
| | Run As | , | 1 Android Application | |
| | Debug As | , | 2 Android 1 Init Test | |
| | Profile As | , | | |
| æ. | Validate | | 4 Java Application Alt+Shift+Y 1 | |
| E- | Team Company 1100 | | Ju 5 11 loit Test Alt+Shift+X T | |
| 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | Compare with Restore from Local History | , | | |
| - <u>a</u> | Android Tools | , | Run Configurations | |
| - 🔂 i | ProDT | , | Advanced | |
| 🗎 (| Properties | Alt-Enter | 🛄 Other | |
| | Resource Configurations | - HILFLINGT | Custom & Library Views | |
| 1 | | | 🔚 Graphical Layout | |
| HolaMundo | | | | |

Nos mostrara un ventana donde elegiremos nuestro dispositivo que será en el cual ejecutaremos nuestra aplicación:

| 😡 Java - HolaMundo/res/layout/activity_main.xn | I - ADT | | | | | | | | | | | |
|---|------------|------------|--------------------------|----------------|-----------|----------|---------------------------------------|-----------------|---------|------|-----------------|---------|
| File Edit Refactor Source Navigate Search Proje | t Run Win | low Help | | | | | | | | | | |
| 📬 • 🗒 🖻 👘 🖡 | 🗹 🕶 🛃 | >> 参 | - 🜔 - 💁 - (| 😫 😌 🕈 🎾 / | 🖇 • 🖄 • 🦮 | | $\models \bullet \Rightarrow \bullet$ | 2 | | | | Qu |
| 📙 Package Explorer 🛛 📄 🤹 🏸 🗖 | 🗆 🛛 🔂 acti | ity, 🚷 Ai | ndroid Device Cho | oser | | | | | × | 1 | - 0 | 📴 Outl |
| 😐 😥 GooAsistenteCliente | ▲ = | P Selec | t a device with min A | VPI level 8. | | | | | | - | | B-14 |
| GooNotification | 🚯 Pak | tte 🖲 C | hoose a running And | froid device | | | | | | 1 | | L |
| E GooServer | 🗁 Fo | m | Serial Number | | AVD Name | | Target | Debug | State | | \odot \odot | |
| HolaMundo | TextView | ar | huawei-um84 | 0-404D8E68AE75 | N/A | | 🖌 2.2.2 | | Online | | | |
| E grow example bolamundo | Puttor | | | | | | | | | | - | |
| MainActivity, java | Dutto | | | | | | | | | | | |
| 🖻 🕮 gen [Generated Java Files] | Che Che | kBo | | | | | | | | | | |
| 🖻 🌐 com.example.holamundo | Checked | ext | | | | | | | | | | |
| 🔅 🚺 BuildConfig.java | Spinn | er C La | , aunch a new Androic | Virtual Device | | | | | | | | |
| 🗄 🔄 R.java | 🗀 Te | et F | | | | | | 1 | | | | |
| E Android 4.2.2 | 🗀 La | ou | AVD Name | Target Name | P | Platform | APILevel | CPU/ABI | Details | | | |
| Android Dependencies | Co | np | Emulauor | Anurola 2.2 | 2 | | 0 | ARM (driftedul) | Start | | | |
| F hin | 🗀 In | adi | | | | | | | | | | |
| 🗄 😓 libs | C Tir | | | | | | | | | | | |
| android-support-v4.jar | | ie i | | | | | | | | | | |
| 🖻 😓 res | | ins | | | | | | | | | | |
| 🗈 🧽 drawable-hdpi | Ad | /ar | | | | | | | Refresh | | | 🔲 Proj |
| 🗁 drawable-ldpi | 🗀 Ot | ier | | | | | | | | | - | Id |
| 🖽 🗁 drawable-mdpi | Custo | 18 | 1 | | | | | | Manager | | Þ | E Laye |
| E C drawable-xhdpi | 🗕 🔳 Gra | ohic | | | | | | | 1 | | | Pade |
| ic launcher.png | | - 0 | se same device for f | ucure launches | | | | OK | Cancel | | | Conl |
| E 🗁 layout | Con | ole | | | | | | | | 1- 🚺 | • • • | 🗦 Relat |
| | DDMS | | | | | | | | | | | Gr |
| h 👝 | | | | | | | | | | | ▲ | - 1g |

Seguido pulsamos ok y se instalara la aplicación en nuestro dispositivo





<u>Cookbook:</u> Comenzando a Programar en Android

ALONSO PARRA Alienx9889 * CESAR VIELMA FREDDY RONDON cesarvielma * spantons * JOSE MARQUEZ joseangel2212

* @gmail.com

Universidad de Los Andes Escuela de Ingeniería de Sistemas La Hechicera, Mérida 5101

RESUMEN: En este artículo se explicara un poco como está estructurado el sistema Android para Desarrolladores. Se encontrará información relevante sobre la interfaz de usuario de este sistema operativo. **PALABRAS CLAVES:** Android, Interfaz, Estructura.

ABSTRACT: This article will explain a bit like Android system is structured for Developers. You will find relevant information about the user interface of the operating system. *Keywords:* Android, Interface, Structure.

I. INTRODUCCION

Hace unos años, Google decidió que debía expandir su negocio hacia los móviles y que mejor estrategia que crear un sistema operativo móvil propio, gratis y con varios de los mas grandes fabricantes de celulares como respaldo. Asi nace **Android**, un sistema operativo móvil open source, basado en Linux, que revolucionó el mercado de los smartphones e inició una carrera tecnológica que continúa al día de hoy.

Android es un sistema operativo que puede ser adoptado por cualquier fabricante de celulares – aunque existe un consorcio de los fabricantes más importantes – y permite realizar tareas que se asemejan a una PC, como navegar la web, leer emails, descargar aplicaciones, etc.

1. DESCRIPCION: ESTRUCTURA DE UNA APLICACIÓN ANDROID

Una aplicación Android está compuesta por varias funciones. Ejemplos como reproducir un archivo de música, envío y recepción de SMS, llamadas a contactos de nuestra agenda. Estas funciones están clasificadas en los siguientes componentes:

- Activities
- Services
- Content Providers

1.1 ACTIVITIES

Una actividad es un componente de aplicación que proporciona una pantalla con la que los usuarios pueden interactuar con el fin de hacer algo, como marcar el teléfono, tomar una foto, envía un correo electrónico, o ver un mapa. La ventana normalmente llena la pantalla, pero puede ser menor que la pantalla y flotan en la parte superior de otras ventanas.

Las actividades a su vez están compuestas por componentes tales como:

- Fragments
- Loader
- Tasks

1.2 SERVICIES

Un servicio es un componente de aplicación que puede realizar operaciones de larga duración en el fondo y no proporciona una interfaz de usuario. Se puede iniciar un servicio y continuará funcionando en segundo plano, incluso si el usuario cambia a otra aplicación. Además, un componente puede unirse a un servicio para interactuar con él e incluso realizar la comunicación entre procesos (IPC). Por ejemplo, un servicio puede manejar las transacciones de red, reproducir música, realizar operaciones con archivos de E / S, o interactuar con un proveedor de contenidos, todo desde el fondo.

1.3 CONTENT PROVIDERS

Los Content Providers gestionan el acceso a un conjunto estructurado de datos. Ellos encapsulan los datos, y proporcionan mecanismos para la definición de seguridad de los datos. Los proveedores de contenidos son la interfaz estándar que conecta los datos en un solo proceso con el código que se ejecuta en otro proceso.

2. DESCRIPCION: INTERFAZ DE USUARIO

La Interfaz de usuario de una aplicación Android es todo lo que el usuario puede ver e interactuar. Ofrece una variedad de componentes de interfaz de usuario, tales como objetos de diseño estructurados y controles de interfaz de usuario que le permiten construir la interfaz gráfica de usuario para un proyecto.

En la interfaz de usuario contamos con varios componentes que hacen funciones importantes en nuestro proyecto Android.

2.1 LAYOUTS

Un Layout define la estructura visual para una interfaz de usuario, tales como la interfaz de usuario para un widget de actividad o aplicación. Se puede declarar un diseño de dos maneras:

• Declarar elementos de la interfaz en XML.

Android proporciona un vocabulario XML directo que corresponde a las clases y subclases Ver, tales como aquellos para los widgets y diseños.

• Crear instancias elementos de diseño en tiempo de ejecución.

Su aplicación puede crear vistas y objetos ViewGroup (y manipular sus propiedades) mediante programación.

A continuación observaremos un ejemplo de cómo implementar un sencillo layout usando XML.



Figura 1. Layout en formato XML

2.2 DIALOGS

Un diálogo es una pequeña ventana que solicita al usuario que tome una decisión o introduzca información adicional. Un diálogo no ocupa toda la pantalla y se utiliza normalmente para los eventos modales que requieren los usuarios para tomar una acción antes de que puedan proceder.



Figura 2. Dialogo en Android

2.3 NOTIFICATIONS

Una notificación es un mensaje que se puede mostrar al usuario fuera de la interfaz de usuario normal de su aplicación. Cuando usted le dice al sistema para emitir una notificación, primero aparece como un icono en el área de notificación. Para ver los detalles de la notificación, el usuario abre el buzón de notificaciones. Tanto el área de notificación y el buzón de notificaciones son áreas controlados por el sistema que el usuario puede ver en cualquier momento.



Figura 3. Ejemplo Notificación en Android 2.4 TOASTS

Un Toast proporciona información sencilla acerca de una operación en una pequeña ventana emergente. Se llena sólo la cantidad de espacio requerido para el mensaje y la actividad actual permanece visible e interactiva. Por ejemplo, navegar fuera de un correo electrónico antes de enviar se dispara un "Proyecto guardado" para hacerle saber que se puede continuar con la edición posterior. Los Toasts desaparecen automáticamente después de un tiempo de espera.

| | | ** 35 | ul. | 9 | 9:48 |
|-----------------------------------|------------|-------|-----|---|------|
| Android Test Hello World, Mair | Test Toast | | | | |
| | | | | | |

Figura 3. Ejemplo Notificación en Android