

Android Debug Bridge: ADB

Sumario

- 1 Introducción
- 2 Dispositivos conectados
- 3 O shell do dispositivo
 - ♦ 3.1 Xestión dunha BBDD con sqlite3
- 4 Sacar ficheiros do dispositivo
- 5 Instalar unha aplicación
- 6 Introducir ficheiro no dispositivo
- 7 Desinstalar unha aplicación do dispositivo
- 8 Varios dispositivos
- 9 Conectar un dispositivo físico
 - ♦ 9.1 Conectar un dispositivo físico en Windows

Introdución

- O **Android Debug Bridge ADB** é unha utilidade que permite comunicarse cun dispositivo virtual ou real.
- Esta composto dun servidor que sempre está funcionando no ordenador real e lánzase cando se inicia Eclipse ou cando se executa **adb start-server**. Este servidor sempre escoita nun porto impar comezando no 5555.
- O comando atópase no cartafol do SDK en **platform-tools/adb**.
- Funciona do mesmo xeito nos tres SOs.
- Máis información en: <http://developer.android.com/tools/help/adb.html>

Dispositivos conectados

- Dispositivos conectados

```
ladmin@ubase: -
ladmin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb
Android Debug Bridge version 1.0.31

  -a                    - directs adb to listen on all interfaces for a connection
  -d                    - directs command to the only connected USB device
                        - returns an error if more than one USB device is present.
  -e                    - directs command to the only running emulator.
                        - returns an error if more than one emulator is running.
  -s <specific device> - directs command to the device or emulator with the given
                        - serial number or qualifier. Overrides ANDROID_SERIAL
                        - environment variable.
  -p <product name or path> - simple product name like 'sooner', or
                        - a relative/absolute path to a product
                        - out directory like 'out/target/product/sooner'.
                        - If -p is not specified, the ANDROID_PRODUCT_OUT
                        - environment variable is used, which must
                        - be an absolute path.
  -H                    - Name of adb server host (default: localhost)
  -P                    - Port of adb server (default: 5037)
  devices [-l]          - list all connected devices
                        - ('-l' will also list device qualifiers)
  connect <host>[:<port>] - connect to a device via TCP/IP
                        - Port 5555 is used by default if no port number is specified.
  disconnect [<host>[:<port>]] - disconnect from a TCP/IP device.
                        - Port 5555 is used by default if no port number is specified.
                        - Using this command with no additional arguments
                        - will disconnect from all connected TCP/IP devices.

device commands:
  adb push <local> <remote> - copy file/dlr to device
  adb pull <remote> [<local>] - copy file/dlr from device
  adb sync [ <directory> ] - copy host->device only if changed
                        - ('-l' means list but don't copy)
                        - (see 'adb help all')
  adb shell [ ]            - run remote shell interactively
  adb shell <command>      - run remote shell command
  adb emu <command>        - run emulator console command
  adb logcat [ <filter-spec> ] - View device log
  adb forward --list       - list all forward socket connections.
```

A execución do comando `<ruta sdk>/platform-tools/adb` amosa a axuda do mesmo.

```
ladmin@ubase: ~
ladmin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb devices
List of devices attached
emulator-5554    device

ladmin@ubase:~$
ladmin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb devices -l
List of devices attached
emulator-5554    device product:sdksdk_x86 model:Android_SDK_built_for_x86 device:generic_x86

ladmin@ubase:~$
```

`<ruta sdk>/platform-tools/adb devices` amosa os dispositivos reais e virtuais conectados ao ordenador. Co parámetro `-l` amosa o tipo de dispositivo.

O shell do dispositivo

- Permite executar comandos dentro do dispositivo android.
- Android está baseado en Linux, por tanto poderanse executar moitos dos comandos dese SO nun dispositivo android.
- **Importante:** Para acceder en modo root en AVDs con versións de android 7 ou superior débese executar antes de entrar nel: **adb root**.

- `adb shell`

```
ladmin@ubase: ~
ladmin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb shell
root@generic_x86:/ #
```

`<ruta sdk>/platform-tools/adb shell` permite conectarse á consola do dispositivo. Ollo que neste exemplo hai un só dispositivo conectado, logo veremos como se actúa cando hai dous ou máis.

```
ladmin@ubase: ~
ladmin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb shell
root@generic_x86:/ #
root@generic_x86:/ # pwd
/
root@generic_x86:/ #
```

A consola do dispositivo e a execución do comando `pwd`.

```
ladmin@ubase: ~
root@generic_x86:/ # df
Filesystem      Size      Used      Free    Blksize
/dev            249.5M    84.0K    249.4M    4096
/mnt/secure     249.5M     0.0K    249.5M    4096
/mnt/asec       249.5M     0.0K    249.5M    4096
/mnt/obb        249.5M     0.0K    249.5M    4096
/system         287.1M    287.1M     0.0K    4096
/data           194.0M    11.7M    182.3M    4096
/cache          64.0M     1.1M    62.9M    4096
/storage/sdcard 126.0M     8.5K    126.0M    512
/mnt/secure/asec 126.0M     8.5K    126.0M    512
root@generic_x86:/ #
```

A execución do comando `df` para ver os puntos de montaxe, os seus tamaños e consumos de espazo.

```
ladmin@ubase: ~
root@generic_x86:/ # mount
rootfs / rootfs ro,relatime 0 0
tmpfs /dev tmpfs rw,seclabel,nosuid,relatime,nodev 755 0 0
devpts /dev/pts devpts rw,seclabel,relatime,nodev 0 0 0
proc /proc proc rw,relatime 0 0
sysfs /sys sysfs rw,seclabel,relatime 0 0
selinuxfs /sys/fs/selinux selinuxfs rw,relatime 0 0
debugfs /sys/kernel/debug debugfs rw,relatime 0 0
tmpfs /mnt/secure tmpfs rw,seclabel,relatime,nodev 700 0 0
tmpfs /mnt/asec tmpfs rw,seclabel,relatime,nodev 755,gid=1000 0 0
tmpfs /mnt/obb tmpfs rw,seclabel,relatime,nodev 755,gid=1000 0 0
/dev/block/mtdblock0 /system yaffs2 ro,seclabel,relatime 0 0
/dev/block/mtdblock1 /data yaffs2 rw,seclabel,nodev,relatime 0 0
/dev/block/mtdblock2 /cache yaffs2 rw,seclabel,nosuid,nodev,relatime 0 0
/dev/block/vold/179:0 /storage/sdcard vfat rw,direct,nosuid,nodev,noexec,relatime,uid=1000,gid=1015,fnmask=0702,dnmask=0702,allow_utime=0020,codepage=cp437,iocharset=iso8859-1,shortname=mixed,utf8,errors=remount-ro 0 0
tmpfs /storage/sdcard/.android_secure tmpfs ro,seclabel,relatime,size=0k,nodev 0 0
root@generic_x86:/ #
```

A execução do comando **mount** que amosa en que directorios están montado os dispositivos físicos. Observar onde está montada a tarxeta SD, o formato de ficheiros, a codificación de caracteres, etc.

```
admin@ubase: ~
root@generic_x86:/ # ls -l
drwxr-xr-x root root 2013-10-26 03:25 acct
drwxrwx--- system cache 2013-10-25 11:49 cache
dr-x----- root root 2013-10-26 03:25 config
lrwxrwxrwx root root 2013-10-26 03:25 d -> /sys/kernel/debug
drwxrwx--x system system 2013-10-25 13:10 data
-rw-r--r-- root root 116 1969-12-31 19:00 default.prop
drwxr-xr-x root root 2013-10-26 03:25 dev
lrwxrwxrwx root root 2013-10-26 03:25 etc -> /system/etc
-rw-r--r-- root root 8753 1969-12-31 19:00 file_contexts
-rw-r----- root root 495 1969-12-31 19:00 fstab.goldfish
-rwxr-x--- root root 359680 1969-12-31 19:00 init
-rwxr-x--- root root 2660 1969-12-31 19:00 init.goldfish.rc
-rwxr-x--- root root 19930 1969-12-31 19:00 init.rc
-rwxr-x--- root root 1795 1969-12-31 19:00 init.trace.rc
-rwxr-x--- root root 3915 1969-12-31 19:00 init.usb.rc
drwxrwxr-x root system 2013-10-26 03:25 mnt
dr-xr-xr-x root root 2013-10-26 03:25 proc
-rw-r--r-- root root 2109 1969-12-31 19:00 property_contexts
drwx----- root root 2013-08-01 05:04 root
drwxr-x--- root root 1969-12-31 19:00/sbin
lrwxrwxrwx root root 2013-10-26 03:25 sdcard -> /storage/sdcard
-rw-r--r-- root root 611 1969-12-31 19:00 seapp_contexts
-rw-r--r-- root root 63747 1969-12-31 19:00 sepolicy
d---r-x--- root sdcard_r 2013-10-26 03:25 storage
dr-xr-xr-x root root 2013-10-26 03:25 sys
drwxr-xr-x root root 2013-08-01 04:31 system
-rw-r--r-- root root 272 1969-12-31 19:00 ueventd.goldfish.rc
-rw-r--r-- root root 4824 1969-12-31 19:00 ueventd.rc
lrwxrwxrwx root root 2013-10-26 03:25 vendor -> /system/vendor
root@generic_x86:/ #
```

ls -l amosa o contido en formato lista do directorio actual: o raíz. Na carpeta **data** ...

```
admin@ubase: ~
root@generic_x86:/ # cd data
root@generic_x86:/data #
root@generic_x86:/data # ls
app
app-asec
app-lib
app-private
backup
bugreports
dalvik-cache
data
dontpanic
drm
local
lost+found
media
mediadrn
misc
nativebenchmark
nativetest
property
resource-cache
security
ssh
system
user
root@generic_x86:/data #
```

... está outra carpeta chamada **data** que ...

```
admin@ubase: ~
root@generic_x86:/data # cd data
root@generic_x86:/data/data # ls
com.android.backupconfirm
com.android.browser
com.android.calculator2
com.android.calendar
com.android.camera
com.android.certinstaller
com.android.contacts
com.android.customlocale2
com.android.defcontainer
com.android.deskclock
com.android.development
com.android.development_settings
com.android.dialer
com.android.dreams.basic
com.android.email
com.android.emulator.connectivity.test
com.android.emulator.gps.test
com.android.exchange
com.android.fallback
com.android.gallery
com.android.gesture.builder
com.android.htmlviewer
com.android.inputdevices
com.android.inputmethod.latin
com.android.inputmethod.pinyin
com.android.keychain
com.android.launcher
com.android.location.fused
com.android.nms
com.android.music
com.android.packageinstaller
com.android.phone
com.android.protips
com.android.providers.applications
com.android.providers.calendar
com.android.providers.contacts
com.android.providers.downloads
```

... contén as aplicacións instaladas no dispositivo. Entre elas unha chamada **com.android.providers.contacts**.

Xestión dunha BBDD con sqlite3

- A modo de exemplo vaise ver como facer consultas básicas, dentro do dispositivo, co xestor **sqlite3**.
- Máis información en: <http://www.sqlite.org/>
- Vaise traballar coa bases de datos dos contactos do teléfono.

- **sqlite3**

```
admin@ubase: ~
root@generic_x86:/data/data # cd com.android.providers.contacts
root@generic_x86:/data/data/com.android.providers.contacts #
root@generic_x86:/data/data/com.android.providers.contacts # ls
cache
databases
files
lib
shared_prefs
```

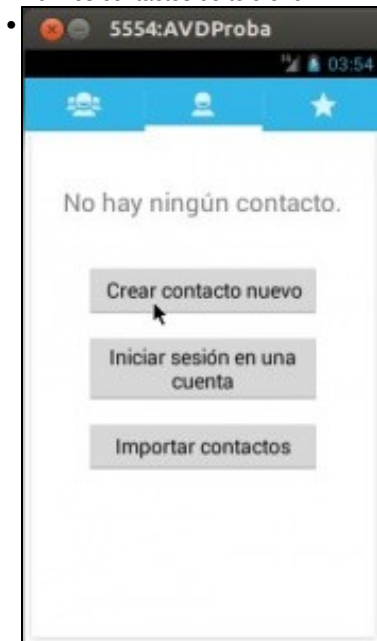
Situarse na carpeta da aplicación. Toda aplicación ten unha estrutura de ficheiros semellante. Hai un directorio chamado **databases**.

```
admin@ubase: ~
root@generic_x86:/data/data/com.android.providers.contacts # cd databases/
root@generic_x86:/data/data/com.android.providers.contacts/databases # ls
contacts2.db
contacts2.db-journal
profile.db
profile.db-journal
```

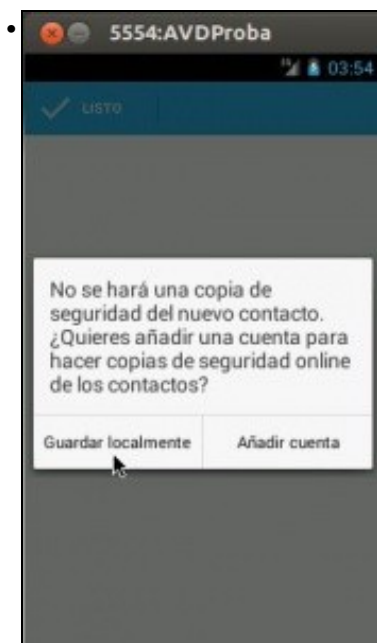
Contén un ficheiro **sqlite3** que é unha base de datos chamada **contacts2.db**.



Abrir os contactos do teléfono.



Crear un novo contacto ...



Localmente ...



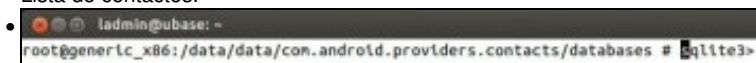
Indicar o nome e o número e gardar o contacto.



Volver á axenda.



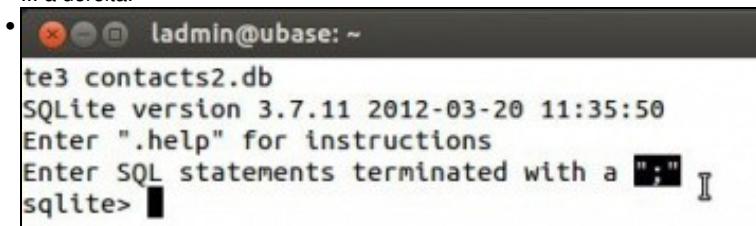
Lista de contactos.



Lanzar o xestor de Base de Datos. **sqlite3 contacts2.db**. Observar que consola se despraza cara ...



... a dereita.



Todo comando empieza por **punto "."**, salvo as sentencias sql, e estas deben rematar en **";"**.

```
admin@ubase:~$ sqlite3
SQLite version 3.7.11 2012-03-28 11:35:50
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .help
.backup ZDB? FILE      Backup DB (default "main") to FILE
.ball ON|OFF          Stop after hitting an error. Default OFF
.databases             List names and files of attached databases
.dump ?TABLE? ...     Dump the database in an SQL text format
                      If TABLE specified, only dump tables matching
                      LIKE pattern TABLE.
.echo ON|OFF          Turn command echo on or off
.exit                 Exit this program
.explain ?ON|OFF?      Turn output mode suitable for EXPLAIN on or off.
                      With no args, it turns EXPLAIN on.
.header(s) ON|OFF     Turn display of headers on or off
.help                 Show this message
.import FILE TABLE   Import data from FILE into TABLE
.indices ?TABLE?      Show names of all indices
                      If TABLE specified, only show indices for tables
                      matching LIKE pattern TABLE.
.log FILE|off         Turn logging on or off. FILE can be stderr/stdout
.mode MODE ?TABLE?    Set output mode where MODE is one of:
                      csv      Comma-separated values
                      column   Left-aligned columns. (See .width)
                      html     HTML <table> code
                      insert   SQL insert statements for TABLE
                      line      One value per line
                      list      Values delimited by .separator string
                      tabs      Tab-separated values
                      tcl       TCL list elements
.nullvalue STRING     Print STRING in place of NULL values
.output FILENAME       Send output to FILENAME
.output stdout         Send output to the screen
.prompt MAIN CONTINUE Replace the standard prompts
.quit                 Exit this program
.read FILENAME         Execute SQL in FILENAME
.restore ZDB? FILE     Restore content of DB (default "main") from FILE
.schema ?TABLE?       Show the CREATE statements
                      If TABLE specified, only show tables matching
```

.help amosa a axuda do xestor.

```
admin@ubase:~$ sqlite3
SQLite version 3.7.11 2012-03-28 11:35:50
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .help
.backup ZDB? FILE      Backup DB (default "main") to FILE
.ball ON|OFF          Stop after hitting an error. Default OFF
.databases             List names and files of attached databases
.dump ?TABLE? ...     Dump the database in an SQL text format
                      If TABLE specified, only dump tables matching
                      LIKE pattern TABLE.
.echo ON|OFF          Turn command echo on or off
.exit                 Exit this program
.explain ?ON|OFF?      Turn output mode suitable for EXPLAIN on or off.
                      With no args, it turns EXPLAIN on.
.header(s) ON|OFF     Turn display of headers on or off
.help                 Show this message
.import FILE TABLE   Import data from FILE into TABLE
.indices ?TABLE?      Show names of all indices
                      If TABLE specified, only show indices for tables
                      matching LIKE pattern TABLE.
.log FILE|off         Turn logging on or off. FILE can be stderr/stdout
.mode MODE ?TABLE?    Set output mode where MODE is one of:
                      csv      Comma-separated values
                      column   Left-aligned columns. (See .width)
                      html     HTML <table> code
                      insert   SQL insert statements for TABLE
                      line      One value per line
                      list      Values delimited by .separator string
                      tabs      Tab-separated values
                      tcl       TCL list elements
.nullvalue STRING     Print STRING in place of NULL values
.output FILENAME       Send output to FILENAME
.output stdout         Send output to the screen
.prompt MAIN CONTINUE Replace the standard prompts
.quit                 Exit this program
.read FILENAME         Execute SQL in FILENAME
.restore ZDB? FILE     Restore content of DB (default "main") from FILE
.schema ?TABLE?       Show the CREATE statements
                      If TABLE specified, only show tables matching
                      LIKE pattern TABLE.
.separator STRING     Change separator used by output mode and .import
.show                 Show the current values for various settings
.stats ON|OFF         Turn stats on or off
.tables ?TABLE?       List names of tables
                      If TABLE specified, only list tables matching
                      LIKE pattern TABLE.
.timeout MS           Try opening locked tables for MS milliseconds
.vfsname ?AUX?        Print the name of the VFS stack
.width NUM1 NUM2 ...  Set column widths for "column" mode
.timer ON|OFF         Turn the CPU timer measurement on or off
sqlite>
```



```
sqlite> .databases
seq name          file
-----
0  main           /data/data/com.android.providers.contacts/databases/contac
sqlite>
```

```
sqlite> .show
      echo: off
      explain: off
      headers: off
      mode: list
nullvalue: ""
      output: stdout
separator: "|"
      stats: off
      width:
sqlite>
```

```
sqlite> .tables
_sync_state          phone_lookup         view_data_usage_stat
_sync_state_metadata photo_files          view_entities
accounts            properties           view_groups
agg_exceptions      raw_contacts         view_raw_contacts
android_metadata    search_index         view_raw_entities
calls               search_index_content view_stream_items
contacts            search_index_docsize view_v1_contact_methods
data                search_index_segdir  view_v1_extensions
data_usage_stat     search_index_segments view_v1_group_membership
default_directory   search_index_stat    view_v1_groups
deleted_contacts    settings             view_v1_organizations
directories           status_updates       view_v1_people
groups              stream_item_photos   view_v1_phones
ninetypes           stream_items          view_v1_photos
name_lookup         v1_settings          visible_contacts
nickname_lookup     view_contacts         voicemail_status
packages            view_data
```

```
sqlite> select * from raw_contacts
...> |
1|1|0|0|2|1|0|1|0|0|0|0|0|AA_Casa|AA_Casa|40||0|AA_Casa|A|1|AA_Casa|A|1|0|0|
sqlite>
```

```
sqlite> select * from view_v1_phones;  
1|1|0|(986) 000-111|1||111000689|AA_Casa|AA_Casa||||0||0|0||||  
sqlite>
```

```

.mode MODE ?TABLE?      Set output mode where MODE is one of:
I      csv              Comma-separated values
      column           Left-aligned columns. (See .width)
      html             HTML <table> code
      insert           SQL insert statements for TABLE
      line            One value per line
      list            Values delimited by .separator string
      tabs            Tab-separated values
      tcl             TCL list elements

```

```

• sqlite> .mode line
sqlite> select * from raw_contacts;
      _id = 1
      account_id = 1
      sourceid =
raw_contact_is_read_only = 0
      version = 2
      dirty = 1
      deleted = 0
      contact_id = 1
      aggregation_node = 0
      aggregation_needed = 0
      custom_ringtone =
      send_to_voicemail = 0
      times_contacted = 0
      last_time_contacted =
      starred = 0
      display_name = AA_Casa
      display_name_alt = AA_Casa
      display_name_source = 40
      phonetic_name =
      phonetic_name_style = 0
      sort_key = AA_Casa
      phonebook_label = A
      phonebook_bucket = 1
      sort_key_alt = AA_Casa
      phonebook_label_alt = A
      phonebook_bucket_alt = 1
      name_verified = 0
      sync1 =
      sync2 =
      sync3 =
      sync4 =
sqlite>

```

.mode line indica que amose cada campo dunha táboa co seu valor nunha liña por campo. **select * from raw_contacts;**

```

• sqlite> .exit
1|root@generic_x86:/data/data/com.android.providers.contacts/databases #

```

.exit serve para saír do xestor e volver á consola do dispositivo.

```

• root@generic_x86:/data/data/com.android.providers.contacts/databases # exit
ladmin@ubase:~$ ls
Descargas  Escritorio  Imaxes  Música  Ubuntu One  workspace
Documentos  examples.desktop  Modelos  Público  Videos
ladmin@ubase:~$

```

Con **exit** saímos da consola do dispositivo e volvemos á consola do ordenador.

```

• ladmin@ubase:~$ ls
Descargas  Escritorio  Imaxes  Música  Ubuntu One  workspace
Documentos  examples.desktop  Modelos  Público  Videos
ladmin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb shell ls
acct
cache
config
d
data
default.prop
dev
etc
file_contexts
fstab.goldfish
init
init.goldfish.rc
init.rc
init.trace.rc
init.usb.rc
mnt
proc
property_contexts
root
sbin
sdcard
seapp_contexts
sepolicy
storage
sys
system
ueventd.goldfish.rc
ueventd.rc
vendor
ladmin@ubase:~$

```

Con **adb shell comando** execútase o comando na consola do dispositivo e a saída amósase na consola do ordenador.

Nota: Se queremos reiniciar o dispositivo teremos que executar, dentro do shell, as seguintes ordes:

- ◊ stop
- ◊ start

Con isto faremos o 'efecto' de reiniciar. Isto é necesario no caso de ter un servizo que se executa cando se acende o dispositivo Android e queiramos comprobar o seu funcionamento.

Sacar ficheiros do dispositivo

- Outra das funcionalidades que nos permite é sacar ficheiros/carpetas do dispositivo.

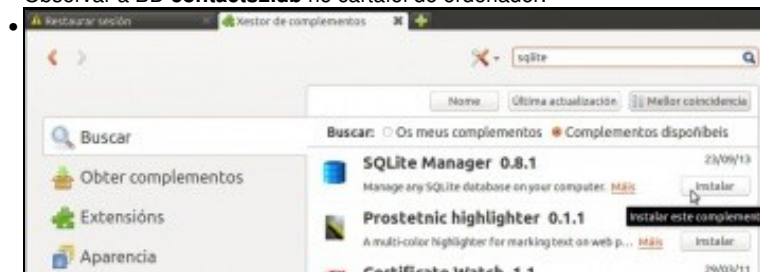
- adb pull

```
ladnín@ubase: ~  
ladnín@ubase:~$ /opt/android-sdk-linux/platform-tools/adb pull /data/data/com.android.providers.contacts/databases/contacts2.db
```

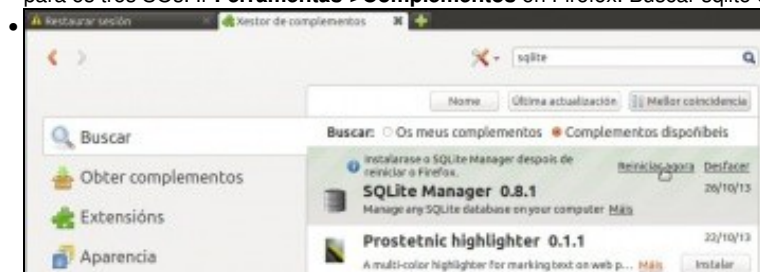
Neste caso sácase do dispositivo a BD anterior de contactos: **contacts2.db**. Co parámetro: **adb pull**. (<ruta sdk>/platform-tools/adb pull /data/data/com.android.providers.contacts/databases/contacts2.db)

```
ladnín@ubase: ~  
ladnín@ubase:~$ /opt/android-sdk-linux/platform-tools/adb pull /data/data/com.android.providers.contacts/databases/contacts2.db  
1263 KB/s (307200 bytes in 0.237s)  
ladnín@ubase:~$ ls  
contacts2.db  Escritorio      Modelos  Ubuntu One  
Descargas    examples.desktop Música    Videos  
Documentos   Imaxes          Público  workspace  
ladnín@ubase:~$
```

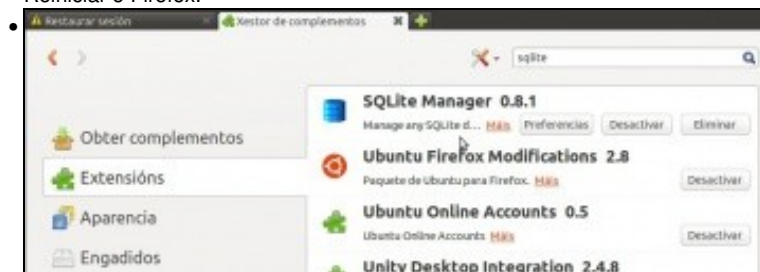
Observar a BD **contacts2.db** no cartafol do ordenador.



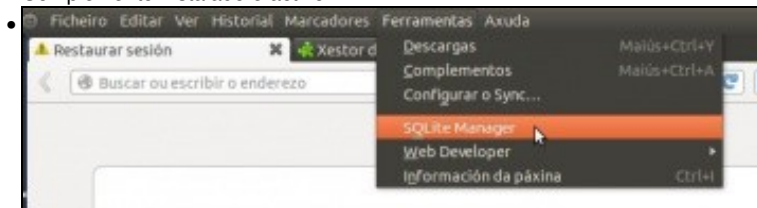
Existen moitos xestores para manipular BBDD sqlite3, neste caso vaise instalar no ordenador un complemento no Mozilla Firefox e así vale para os tres SOs. Ir **Ferramentas->Complementos** en Firefox. Buscar **sqlite** e instalar o complemento **Sqlite Manager**



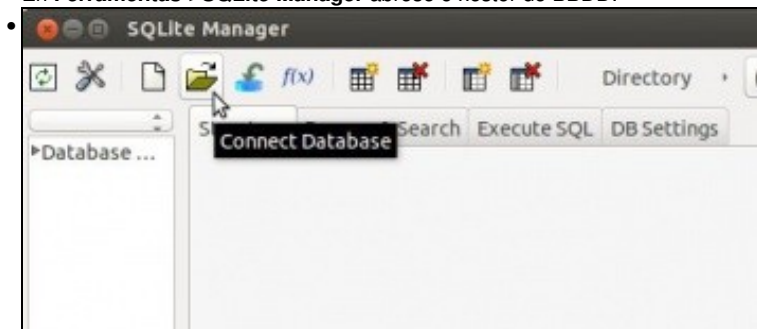
Reiniciar o Firefox.



Complemento instalado e activo.



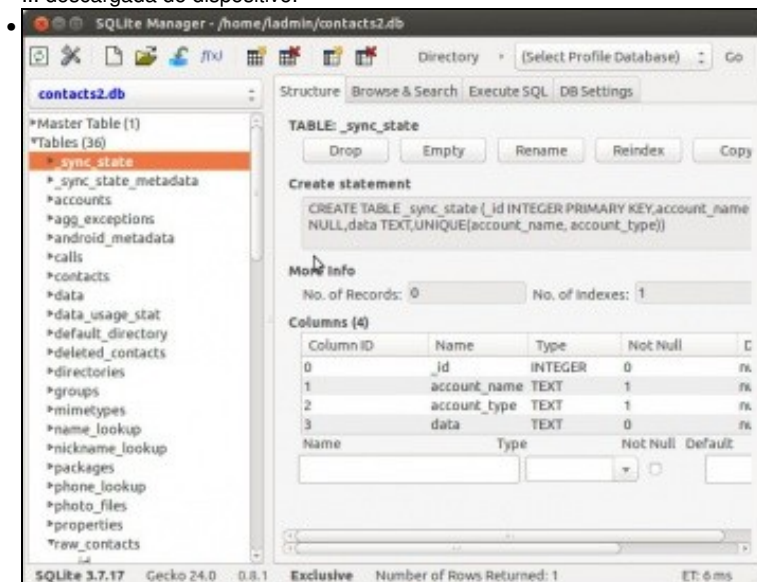
En Ferramentas->SQLite Manager ábrese o xestor de BBDD.



Conectar á BD ...



... descargada do dispositivo.

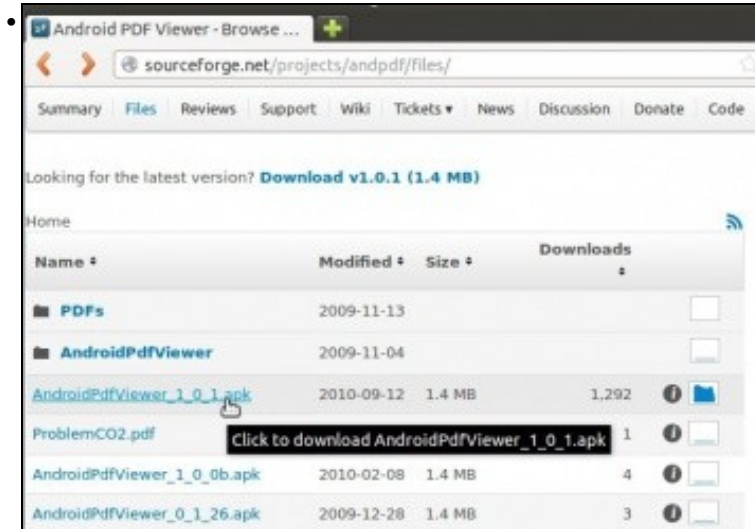


Agora pódese manipular a BD.

Instalar unha aplicación

- As aplicacións instalables teñen extensión **apk** como se verá na UNIDADE 2 do curso: **adb install**.
- Neste caso vaise baixar de internet un visor PDF moi sinxelo para Android.

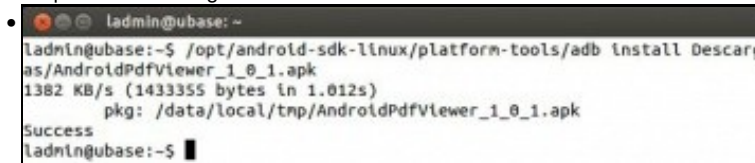
- adb install



Descargar o .apk do visor PDF de <http://sourceforge.net/projects/andpdf/files/>



A aplicación descargada.



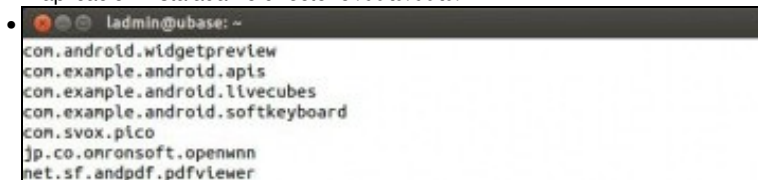
<ruta sdk>/platform-tools/adb install Descargas/AndroidPdfViewer_1_0_1.apk instalará o paquete no dispositivo.



A aplicación instalada.



A aplicación instalada no directorio **/data/data/** ...



... do dispositivo.

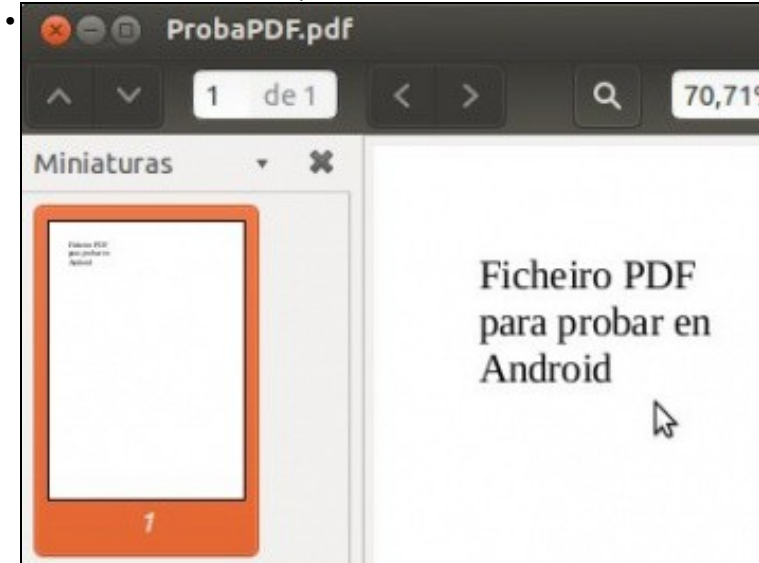
Introducir ficheiro no dispositivo

- Ao igual que se sacan ficheiros do dispositivo tamén se poden introducir ficheiros no mesmo: **adb push**.

- adb push



Crear un documento PDF de proba ou escoller un xa feito.



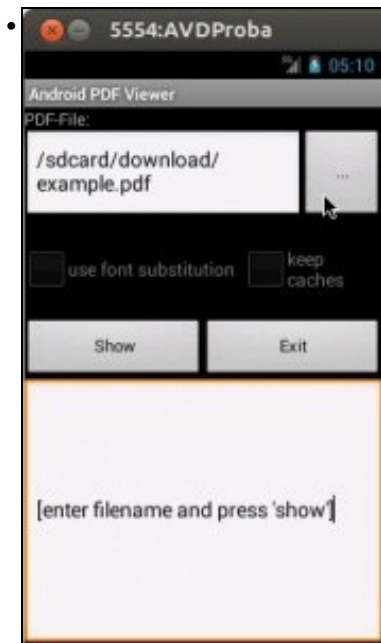
O contido do documento PDF.

-

Imos introducilo no directorio asociado á SD card: **sdcard**

-

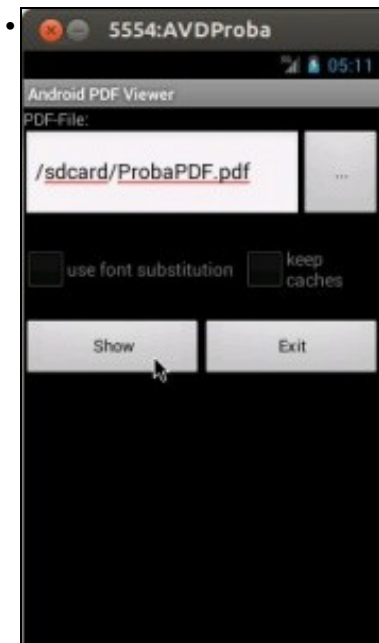
<ruta sdk>/platform-tools/adb push Documentos/ProbaPDF.pdf /sdcard/



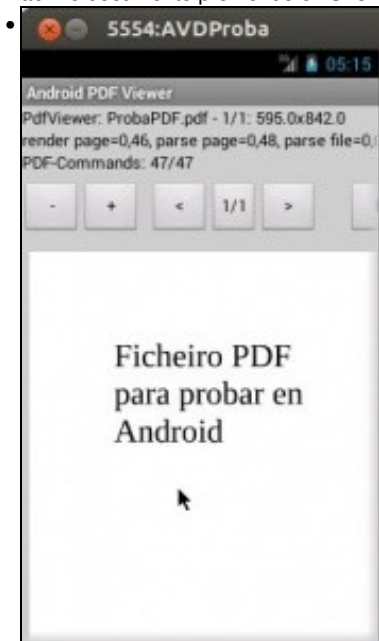
Abrir o visor de PDFs no dispositivo e navegar ...



até /sdcard e



abrir o documento premendo en **show**.



O contido no documento visto no dispositivo.

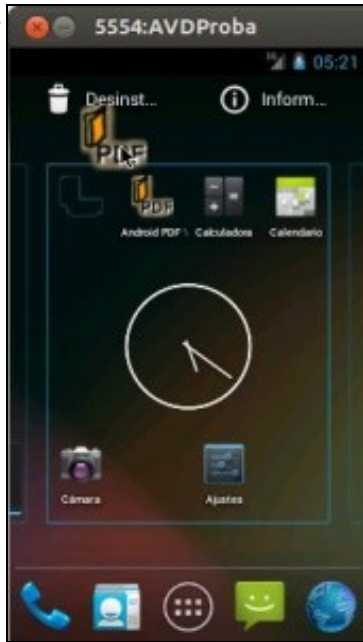


adb shell ls /sdcard amosa que o documento está na tarxeta SD.

Desinstalar unha aplicación do dispositivo

- Ao igual que se pode instalar unha aplicación tamén se pode desinstalar: **adb uninstall**.
- Tamén se pode realizar de modo gráfico.

- adb uninstall



En modo gráfico desinstálase unha aplicación mantendo pulsada e arrastrándoa ao lixo. Vaise realizar con adb.

```
admin@ubase: ~
con.android.widgetpreview
con.example.android.apis
con.example.android.livecubes
con.example.android.softkeyboard
con.svox.pico
jp.co.onronsoft.openwnn
net.sf.andpdf.pdfviewer
```

En /data/data podemos ver o nome da aplicación.

```
admin@ubase: ~
ladningubase:~$ /opt/android-sdk-linux/platform-tools/adb uninstall net.sf.a
ndpdf.pdfviewer
Success
```

<ruta sdk>/platform-tools/adb uninstall net.sf.andpdf.pdfviewer

Varios dispositivos

- Até agora só traballamos con adb cun só dispositivo. Como actuar cando hai máis dun dispositivo real ou virtual?.

- Varios dispositivos



Crear un segundo AVD: **AVDProba2**.



Iniciar os 2 AVDs.

```
ladmin@ubase: ~  
ladmin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb kill-server  
ladmin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb devices  
* daemon not running. starting it now on port 5037 *  
* daemon started successfully *  
List of devices attached  
emulator-5554 device  
emulator-5556 device
```

Parar o servidor adb: **<ruta sdk>/platform-tools/adb kill-server**

```
ladmin@ubase: ~  
ladmin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb devices  
List of devices attached  
emulator-5554 device  
emulator-5556 device  
  
ladmin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb -s emulator-5556 shell ls  
acct  
cache  
config
```

Buscar os dispositivos conectados: **adb devices**. Observar como se inicia o servidor adb.

```
ladmin@ubase: ~  
ladmin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb devices  
List of devices attached  
emulator-5554 device  
emulator-5556 device  
  
ladmin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb -s emulator-5556 shell ls  
acct  
cache  
config
```

Agora, cando hai máis dun dispositivo, para conectarse a un concreto hai que especificalo, sexa este real ou virtual: usando o parámetro **-s nome do dispositivo**. Neste caso **adb -s emulator-5556 shell ls** amosa o contido da raíz do segundo AVD.

Conectar un dispositivo físico

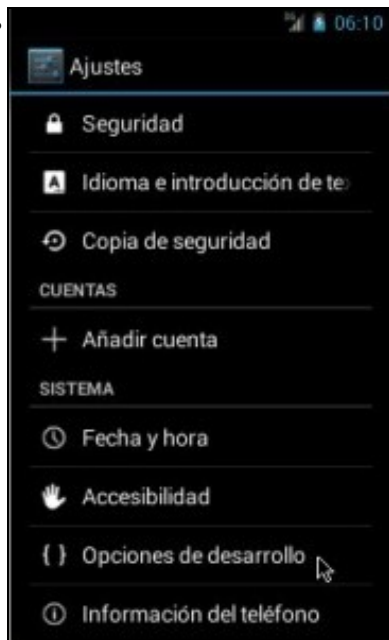
- Agora tócalle a quenda aos dispositivos físicos.
- Hai que habilitar a **Depuración USB**
- Dispositivo físico



Se non aparece en **Ajustes->{ } Opciones de desarrollo**. Ir a **Ajustes -> Información del teléfono**.



Pulsar 7 veces sobre **Número de compilación** e xa aparece ...



... { } **Opciones de desarrollo**. Premer nesa opción. En versións máis antigas ir a **Ajustes-> Aplicaciones->Desarrollo**



Habilitar **Depuración USB**



Aceptar as advertencias.

```
admin@ubase: ~
admin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb devices -l
List of devices attached
emulator-5554      device product:sdk_x86 model:Android_SDK_built_for_x86
emulator-5556      device product:sdk_x86 model:Android_SDK_built_for_x86
20e62b4d           device usb:1-2
```

<ruta sdk>/platform-tools/adb devices -l amosa o nome dos dispositivos e onde está conectado.

```
admin@ubase: ~
admin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb devices
List of devices attached
emulator-5554      device
emulator-5556      device
20e62b4d           device

admin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb -s 20e62b4d shell ls
efs
config
cache
```

<ruta sdk>/platform-tools/adb -s nome-dispositivo é para interactuar con ese dispositivo, neste caso físico.

Conectar un dispositivo físico en Windows

- En Linux e OS X non é preciso instalar o driver do dispositivo para conectarse a el, pero en Windows si.
- Neste caso vaise usar de exemplo un dispositivo físico Samsung.

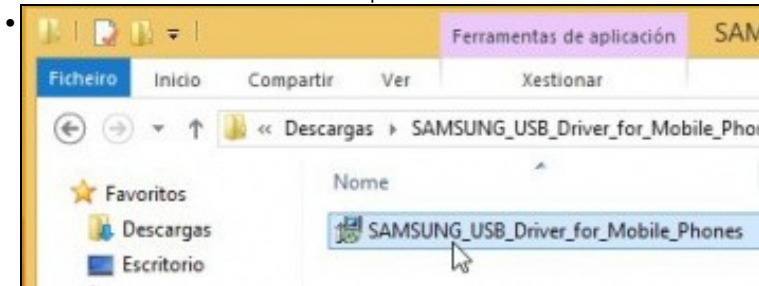
- Dispositivo físico en Windows

```
C:\Windows\system32\cmd.exe
c:\Program Files (x86)\Android\android-sdk\platform-tools>adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
c:\Program Files (x86)\Android\android-sdk\platform-tools>
```

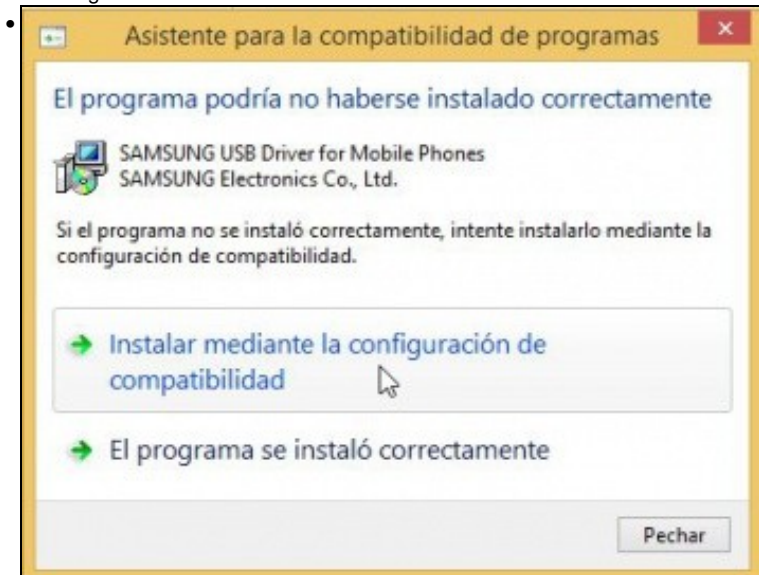
Conectamos o dispositivo físico ao ordenador Windows e <ruta sdk>/platform-tools/adb devices non amosa nada.



Neste caso buscamos o driver do dispositivo.



Descargalo e instalalo.



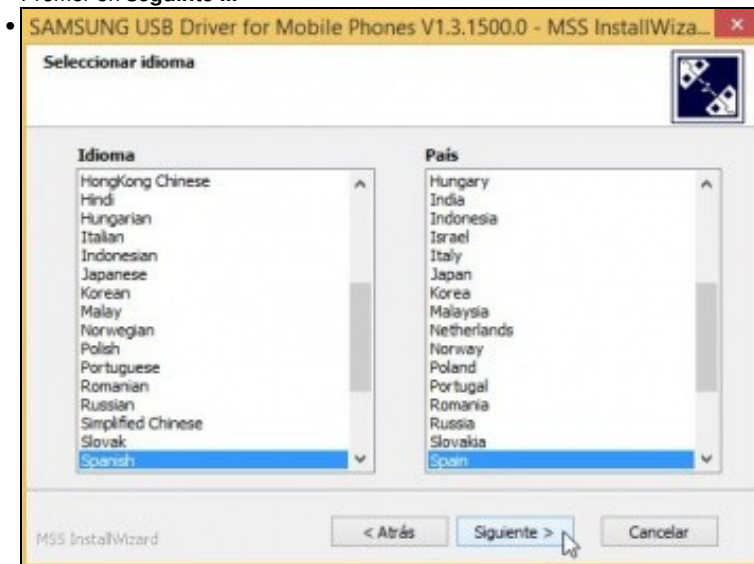
Aceptar as advertencias que amose.



Seguir aceptando ...



Premer en **siguiente** ...



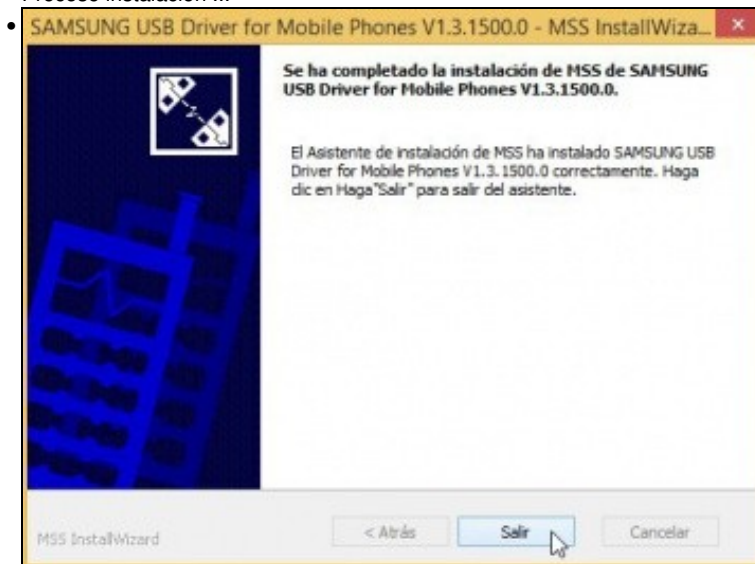
Seleccionar o que indique ...



Instalar ...



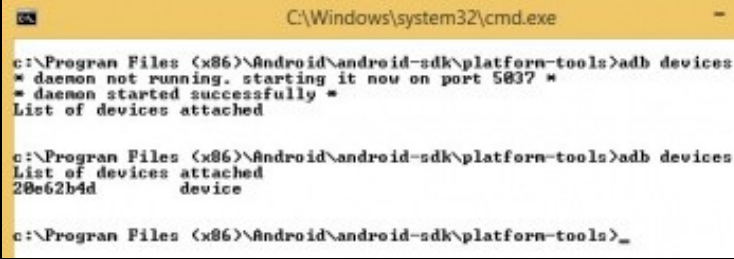
Proceso instalación ...



Rematar instalación.



Aceptar por última vez.

- A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window shows the execution of the 'adb devices' command. The output indicates that the adb daemon was not running and was started successfully on port 5037. It then lists one attached device with ID 20e62b4d.

```
C:\Windows\system32\cmd.exe

c:\Program Files (x86)\Android\android-sdk\platform-tools>adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
20e62b4d    device

c:\Program Files (x86)\Android\android-sdk\platform-tools>
```

<ruta sdk>/platform-tools/adb devices amosa o dispositivo físico conectado ao ordenador.

-- Ángel D. Fernández González e Carlos Carrión Álvarez -- (2020).