

# Fondamenti di sviluppo mobile su Android



Dott. A. Tedeschi  
antonio.tedeschi@uniroma3.it

Corso di  
*Telecomunicazioni Wireless*

a.a. 2016-2017

- Sicurezza Android
  - Reverse engineering apk
  - Proguard offuscatore di codice
- Creazione di apk per la produzione
- Creazione account developer
- Panoramica Developer Console

Android, data la sua diffusione, soffre di problemi di sicurezza

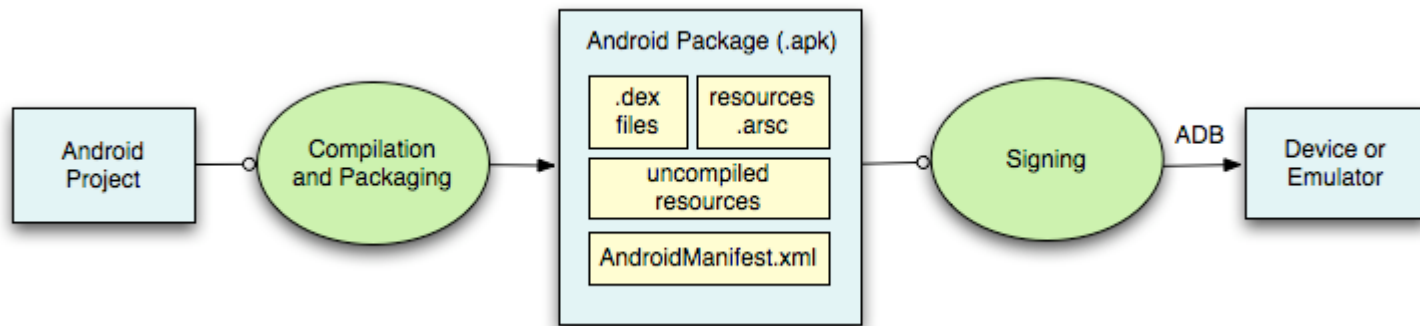
- **sistema operativo**

bug, problemi di crittografia, malware, spyware ecc.  
Problema generalmente risolto dalla Google Mobile  
Division o da antivirus installati sul device

- **applicazioni**

estrazione del codice sorgente e delle relative risorse a  
partire dall'apk  
Problema non risolvibile. Gli sviluppatori possono solo  
rendere la procedura più complicata

Un file .apk (Android Package) contiene tutte le informazioni necessarie per permettere l'esecuzione dell'applicazione creata (sia su dispositivi che su emulatore)

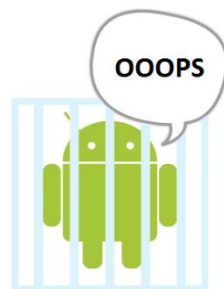




**NOTA!**

La decompilazione (reverse engineering) degli Android Package (apk) non è un'operazione illegale finché non è usata per scopi di pirateria.

Le procedure mostrate sono **solo per scopi didattici**  
Si raccomanda di **NON** utilizzare quanto appreso per scopi illegali in quanto direttamente responsabili.



## Due approcci

- Adozione di tool pronti in rete. Procedura automatica e rapida, necessita di connessione internet, possibilità di essere rintracciati
- Adozione dei seguenti strumenti

- [apk-tool](#)



- [dex2jar](#)

- [JD](#) (Java Decompiler)



Processo di decompilazione più lungo.

- Referenza: <http://www.html.it/articoli/decompilare-ed-offuscare-il-codice-di-unapp-android/>

# Decompilare APK – Primo metodo

Primo metodo: esempio di webapp per decompilare un apk  
<http://www.decompileandroid.com/>

October 16 - We've updated apktool most recent version (2.02)

## Android APK Decompiler

Home | Security Assessments | FAQ | Feedback?

### Decompiling APK files made easy

Choose an APK file and upload it and we'll decompile it in just a couple minutes.

[SELECT FILES](#)

Having trouble with the uploader? Try the [static version](#) instead.

#### What does this do?

All applications for Android phones are distributed as APK Files. These files contain all the code, images and other media necessary to run the application on your phone. This website will decompile the code embedded in APK files and extract all the other assets in the file. For more information check out the [Frequently Asked Questions](#)

#### Don't have an APK file to test?

Here's a [sample APK file](#) that you can use. You can also view the output of a completed extraction [here](#).

Proudly hosted on [Digital Ocean](#)

# Decompilare APK – Primo metodo

October 16 - We've updated apktool most recent version (2.02)

## Android APK Decompiler

[Home](#) | [Security Assessments](#) | [FAQ](#) | [Feedback?](#)

### Decompiling Complete!

Here's a contents of AndroidManifest.xml. You can download the full contents of the APK [here](#)

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="it.antedesk.htmlreverse"
  <application android:allowBackup="true" android:icon="@mipmap/ic_launcher" android:label="@string/app_name"
    <activity android:label="@string/app_name" android:name="it.antedesk.htmlreverseeng.MainActiv
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
  </application>
</manifest>
```

### Did you find this useful?

Sign up to receive occasional email updates on Android development and best security practices!

**Subscribe**

Proudly hosted on [Digital Ocean](#)



# Decompilare APK – Secondo metodo

**dex2jar** è un insieme di file .bat (sotto Windows) e librerie

Per utilizzarlo è necessario

- modificare l'estensione dell'applicazione da decompilare da .apk a .zip
- Estrarre il file **classes.dex**
- Spostare il file classes.dex nella cartella dex2jar contenente l'occorrente per ottenere il codice
- Aprire la finestra di comando dalla cartella dex2jar (**shift+tasto destro-> finestra di comando**)
- Inserire la seguente riga

```
d2j-dex2jar.bat "[PERCORSO_FILE]\classes.dex"
```

Il risultato sarà la creazione di un file **classes\_dex2jar.jar** contenente il codice sorgente dell'app desiderata

## Java Decompiler

Per visualizzare il codice eseguire JD e caricare il file ottenuto dal passo precedente

# Decompilare APK – Secondo metodo

Apktool per funzionare necessita dei seguenti file

- apktool.bat
- apktool\_[Version].jar
- file apk da decompilare

Con sh  
questi

Inserire

Il risult  
basso

esenti

di

```

C:\WINDOWS\system32\cmd.exe
C:\Users\Antedesk\Desktop\Decompilazione\Decompilazione- HTML.it\ApkTool>apktool d "C:\Users\Antedesk\Desktop\Decompilazione\Decompilazione- HTML.it\HTMLReverseEng.apk"
I: Using Apktool 2.0.3 on HTMLReverseEng.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: C:\Users\Antedesk\apktool\framework\1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
C:\Users\Antedesk\Desktop\Decompilazione\Decompilazione- HTML.it\ApkTool>
  
```

Risultato:

- codice smali,
- i file originali codificati,
- le risorse contenute nella cartella res e l'AndroidManifest.xml.

È possibile effettuare il reverse engineering di un file apk  
Possibilità per gli «*hacker*» entrare in possesso di:

- Grafica
- Codice (smali e java)
- Informazioni riservate (es: la chiave delle mappe di Google)

## Soluzione

Utilizzare un offuscatore di codice: [ProGuard](#)

## Permette

- Riduzione del codice. Elimina classi e variabili non utilizzate
- Ottimizzazione del codice e delle risorse (riduce il peso dell'apk)
- Offuscazione del codice tramite la rinominazione di classi, attributi e metodi con nomi semanticamente «oscuri» (non significativi)

Rende più complesso comprendere il codice sorgente

Integrato in Android, non deve essere importato

Necessita di essere abilitato e configurato in base alle esigenze

Viene utilizzato dal compilatore di Android Studio solo se si è in «*release mode*», ossia si crea l'apk che deve essere pubblicato

**Vantaggio:** finché si sta sviluppando l'applicazione (*debug mode*) i possibili errori che si riscontreranno sul dispositivo saranno chiari e leggibili

Per abilitarlo

- modificare la proprietà *minifyEnabled* nel file **build.gradle** cambiando il valore da `false` a `true`.

```
buildTypes {  
    release {  
        minifyEnabled false  
        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'  
    }  
}
```

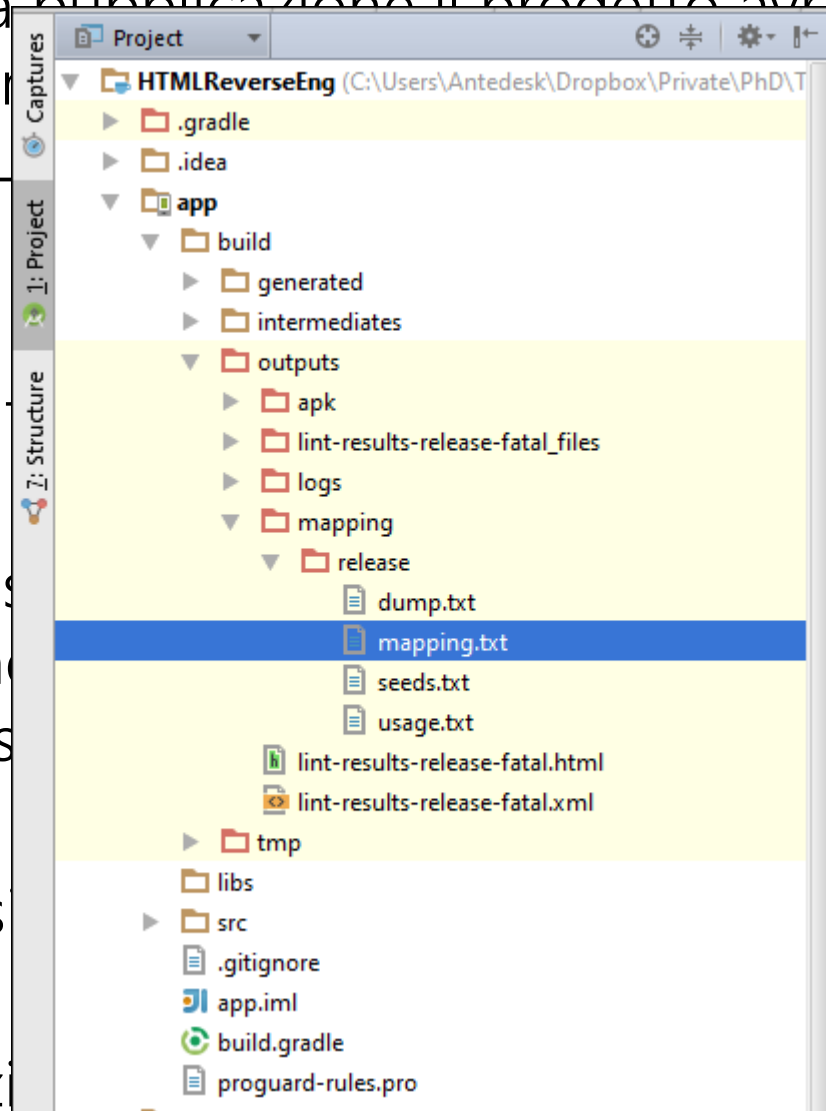
In un progetto Android realizzato su Android Studio è presente il file:

- **proguard-rules.pro**  
Permette di configurare opportunamente Proguard in base alle proprie esigenze definendo regole ad-hoc per l'offuscazione

# Proguard

Creata l'apk per la pubblicazione il progetto avrà una nuova cartella chiamata `outputs` di build. L'output della cartella è:

- **dump.txt**  
Descrive la struttura dell'app
- **mapping.txt**  
Lista delle corrispondenze tra nomi originali e offuscati. Fondamentale per l'analisi dell'app rilasciata
- **seeds.txt**  
Lista delle classi originali
- **usage.txt**  
Lista delle porzioni di codice



, attributi originali  
bug report per

tati offuscati

minati

Non sempre la creazione del file apk va a buon fine generando errori come:

## ClassNotFoundException

La colpa è di Proguard.

Durante la procedura di riduzione – ottimizzazione – offuscamento può andare ad eliminare:

- Una classe referenziata solo nel AndroidManifest.xml
- Un chiamata ad un metodo JNI (definito con l'NDK)
- Referenze dinamiche ad attributi e metodi

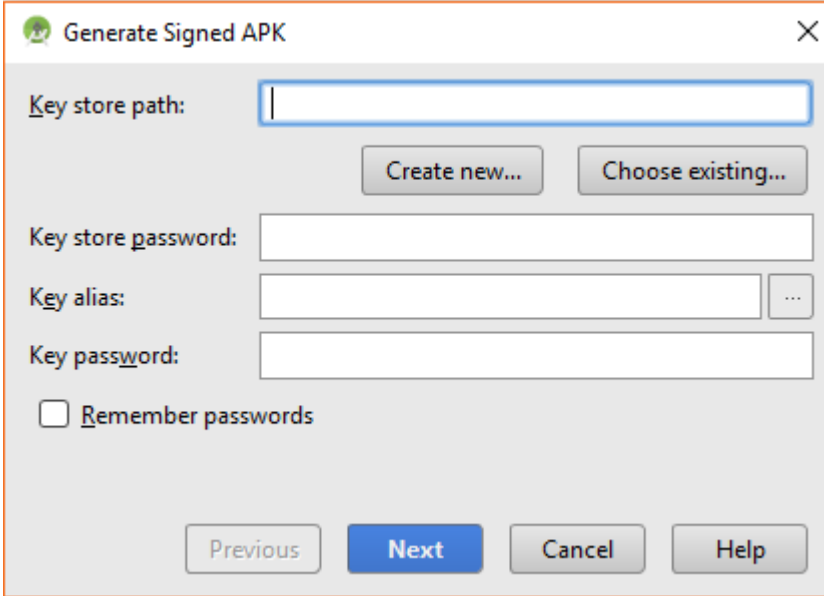
È possibile risolvere il problema aggiungendo al file di configurazione determinate opzioni.

Ad es per risolvere il problema dell'eliminazione di una classe

```
-keep public class <MyClass>
```



Build -> Generate Signed APK



Generate Signed APK

Key store path:

Create new... Choose existing...

Key store password:

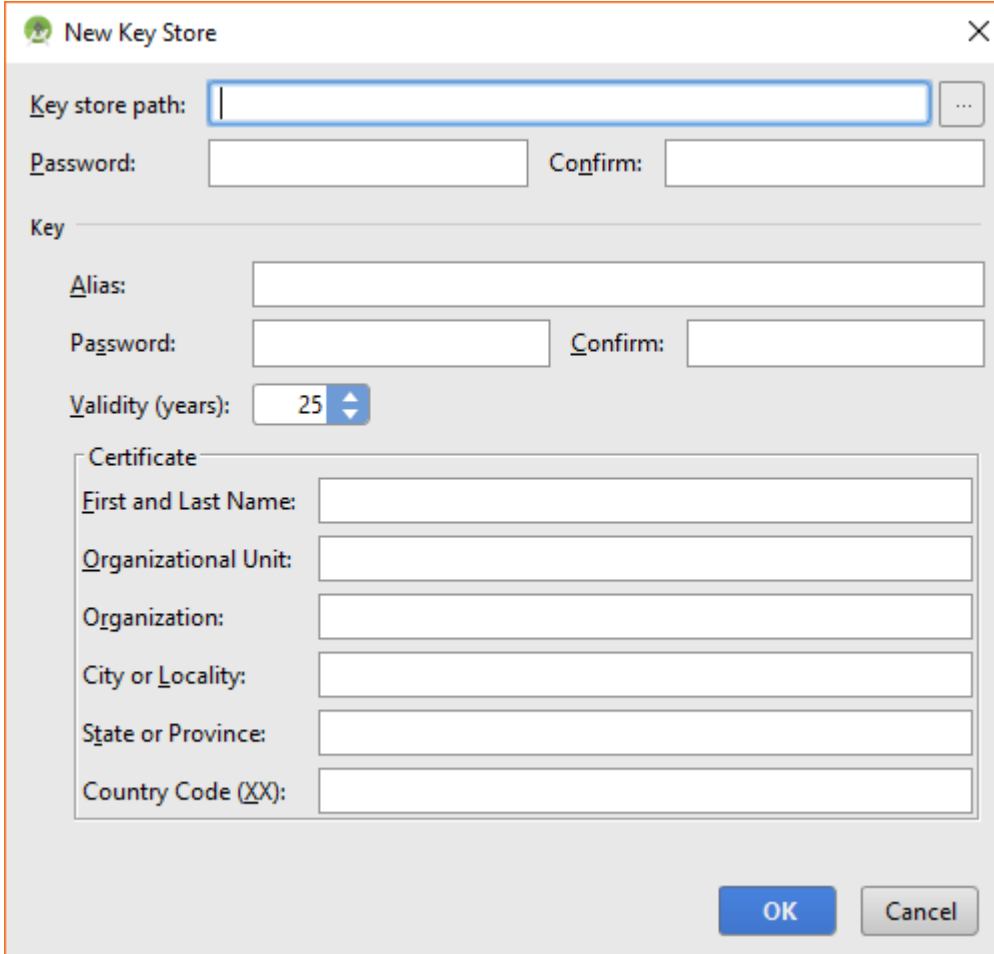
Key alias:  ...

Key password:

Remember passwords

Previous Next Cancel Help

## Creazione di una Key Store per firmare l'apk



**New Key Store**

Key store path:  ...

Password:  Confirm:

Key

Alias:

Password:  Confirm:

Validity (years): 25

Certificate

First and Last Name:

Organizational Unit:

Organization:

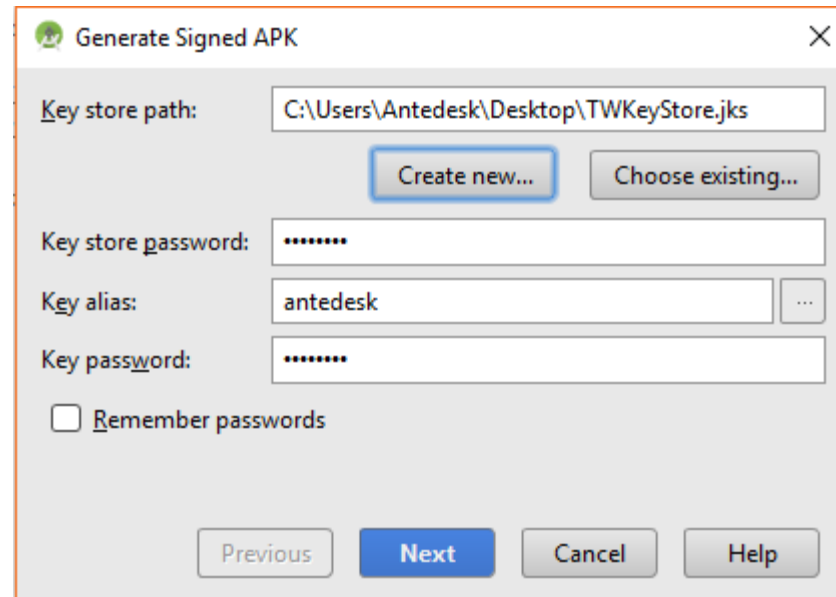
City or Locality:

State or Province:

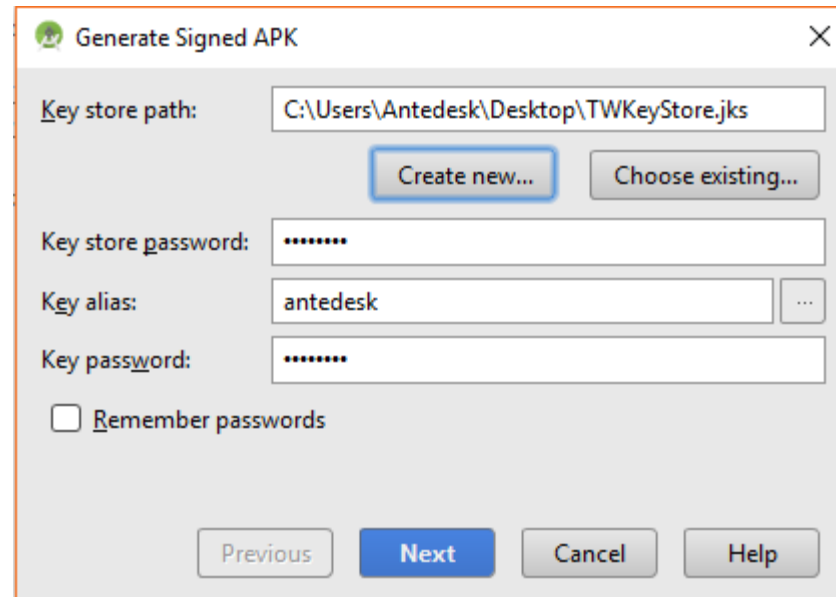
Country Code (XX):

OK Cancel

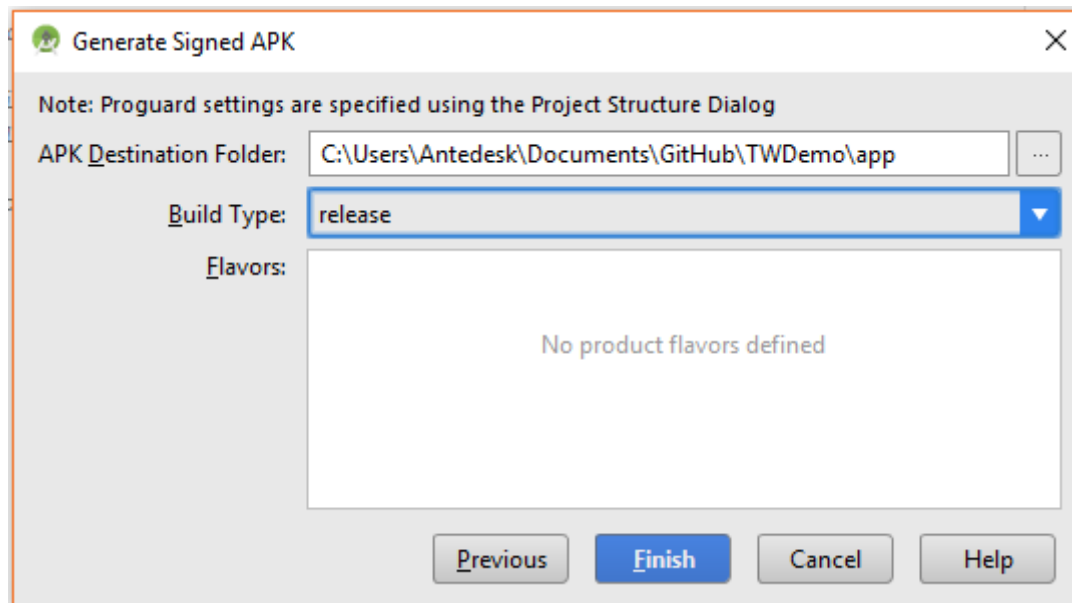
Una volta creata o aggiunta la scheda si auto compilerà



Una volta creata o aggiunta la scheda si auto compilerà



Una volta creata o aggiunta la scheda si auto compilerà



- Creare un account Gmail con il quale accedere al play store
  - Questo account sarà considerato del proprietario per l'account sviluppatore
  - Questo account può rappresentare sia l'azienda che uno sviluppatore
  - Informazioni sul contratto che verrà stretto con Google sono reperibili al seguente link <https://play.google.com/about/developer-distribution-agreement.html>
  - Qualora il nome dell'azienda dovesse essere già preso, suggerisco di utilizzare suffissi come *\_developer*, *\_dev*, *\_team*, simili. In questo modo resta il nome della società seguito da un suffisso che lascia intendere che avete un settore IT
- Collegarsi al sito <https://play.google.com/apps/publish/signup/> accedendo con l'account Gmail appositamente creato

# Creazione account developer

Google play | Developer Console

Accedi con il tuo account Google → Accetta Contratto con gli sviluppatori → Paga la quota di registrazione → Inserisci i dati del tuo account

HAI ESEGUITO L'ACCESSO COME...

Antonio Tedeschi  
ant.tedeschi@gmail.com

Questo è l'account Google che verrà associato alla tua Console per gli sviluppatori.

Se desideri utilizzare un account diverso, puoi scegliere tra le opzioni che seguono. Se rappresenti un'organizzazione, potresti creare un nuovo account Google anziché utilizzare un account personale.

[Accedi con un altro account](#) [Crea un nuovo account Google](#)

PRIMA DI CONTINUARE...

Leggi e accetta il [Contratto di distribuzione per gli sviluppatori di Google Play](#).

**1**  Accetto e sono disposto ad associare la registrazione dell'account al Contratto di distribuzione per gli sviluppatori di Google Play.

Ricontrolla i Paesi in cui puoi distribuire e vendere applicazioni.

Se hai intenzione di vendere app o prodotti in-app, controlla se puoi avere un account commerciante nel tuo Paese.

**2** \$25

Assicurati di avere a portata di mano la tua carta di credito per pagare la quota di registrazione di \$ 25 nel prossimo passaggio.

**3** [Vai al pagamento](#)

(1) Accettare i termini del contratto

(2) Il costo dell'account è di 25\$ circa 19€. L'iscrizione come developer non è annua, pertanto l'importo verrà pagato solo una volta.

(3) Cliccare su vai al pagamento

**NOME E INDIRIZZO**

1

Italia (IT)

Nome

Indirizzo

Codice postale

Città

Agrigento

**METODO DI PAGAMENTO**

Carta di credito o di debito

Numero carta

VISA MASTERCARD AMEX DISCOVER

Data di scadenza

MM / AA

Codice di sicurezza

CVC ?

2

**Indirizzo di fatturazione**

L'indirizzo di fatturazione è uguale a quello inserito in Nome e indirizzo

Inviarmi offerte speciali di Google Wallet, inviti a fornire feedback sui prodotti e newsletter.

Accetto i [Termini di servizio](#) e l'[Informativa sulla privacy](#) di Google Wallet.

Annulla Accetta e continua

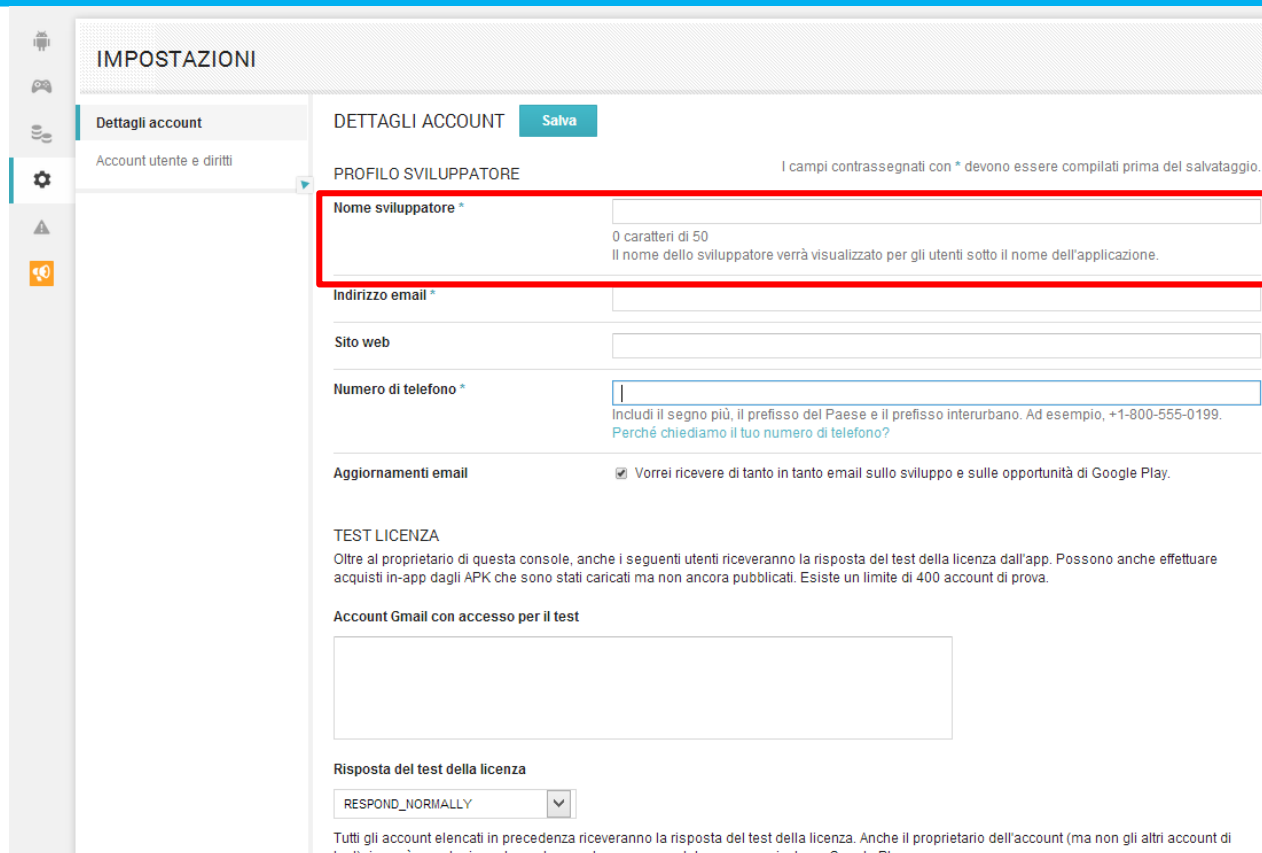
3

(1) A questo punto impostare il proprio profilo aziendale su Google Wallet, riempiendo i campi e inserendo la carta di credito (preferibilmente quella aziendale in quanto sullo stesso profilo di Google Wallet si riceveranno anche i pagamenti dell'applicazione) Informazioni sui metodi di pagamento accettati sono reperibili qui: <https://support.google.com/wallet/answer/105916>

(2) nota: se l'indirizzo di fatturazione è differente da quello inserito. Se così fosse togliere la spunta e compilare la form che verrà mostrata in cui inserire l'indirizzo di fatturazione

(3) Cliccare su accetta e continua





IMPOSTAZIONI

Dettagli account

Account utente e diritti

DETTAGLI ACCOUNT [Salva](#)

PROFILO SVILUPPATORE I campi contrassegnati con \* devono essere compilati prima del salvataggio.

**Nome sviluppatore \***   
0 caratteri di 50  
Il nome dello sviluppatore verrà visualizzato per gli utenti sotto il nome dell'applicazione.

Indirizzo email \*

Sito web

Numero di telefono \*   
Includi il segno più, il prefisso del Paese e il prefisso interurbano. Ad esempio, +1-800-555-0199.  
Perché chiediamo il tuo numero di telefono?

Aggiornamenti email  Vorrei ricevere di tanto in tanto email sullo sviluppo e sulle opportunità di Google Play.

TEST LICENZA  
Oltre al proprietario di questa console, anche i seguenti utenti riceveranno la risposta del test della licenza dall'app. Possono anche effettuare acquisti in-app dagli APK che sono stati caricati ma non ancora pubblicati. Esiste un limite di 400 account di prova.

Account Gmail con accesso per il test

Risposta del test della licenza

Tutti gli account elencati in precedenza riceveranno la risposta del test della licenza. Anche il proprietario dell'account (ma non gli altri account di test) riceverà questa risposta per la app che non sono state ancora caricate su Google Play.

- Effettuato il pagamento si verrà rimandati in un apposita schermata in cui inserire tutte le informazioni inerenti l'account developer incluso il nome sviluppatore e indirizzo mail (suggerisco quello aziendale)
- Il nome sviluppatore è il nome che viene visualizzato sul Google Play Store. Pertanto va scelto con cura se diverso da quello dell'azienda.



Google play | Developer Console

1 Tutte le applicazioni

2 Servizi di giochi

3 Rapporti finanziari

4 Impostazioni

5 Avvisi

6 Comunicazioni

TUTTE LE APPLICAZIONI + Aggiungi nuova applicazione

Filtra

NOME APPLICAZIONE	PREZZO	INSTALLAZIONI CORRENTI/TOTALI
BiciSicura 1.0	Gratuita	

- 1) Main page: sezione in cui si ha accesso alle statistiche rapide delle applicazioni
- 2) Servizi e giochi: se si sviluppano giochi è possibile accedere a questa sezione per impostare alcune utilità
- 3) Rapporti finanziari: permette la gestione dei dati finanziari e dei guadagni delle app a pagamento
- 4) Avvisi
- 5) Comunicazioni Google



In questa sezione è possibile gestire il business delle applicazioni a pagamento.

Cliccando su (1) si verrà rimandati a Google wallet <https://wallet.google.com/> che permetterà di creare un vostro profilo finanziario compilando la form.

Google play | Developer Console

Tutte le applicazioni

Servizi di giochi

Rapporti finanziari

Impostazioni

Avvisi

Comunicazioni

## RAPPORTI FINANZIARI

(Tutti i rapporti vengono forniti come file CSV scaricabili)

Google wallet

Per iniziare a guadagnare con app a pagamento o prodotti in-app, configura un account commerciante Google Wallet.

**1** [Configura subito un account commerciante](#)

[Leggi ulteriori informazioni](#)