ROMA TRE
UNIVERSITÀ DEGLI STUDI

Fondamenti di sviluppo mobile su Android

Dott. A. Tedeschi

antonio.tedeschi@uniroma3.it

Corso di
Telecomunicazioni Wireless

a.a. 2016-2017

Outline

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

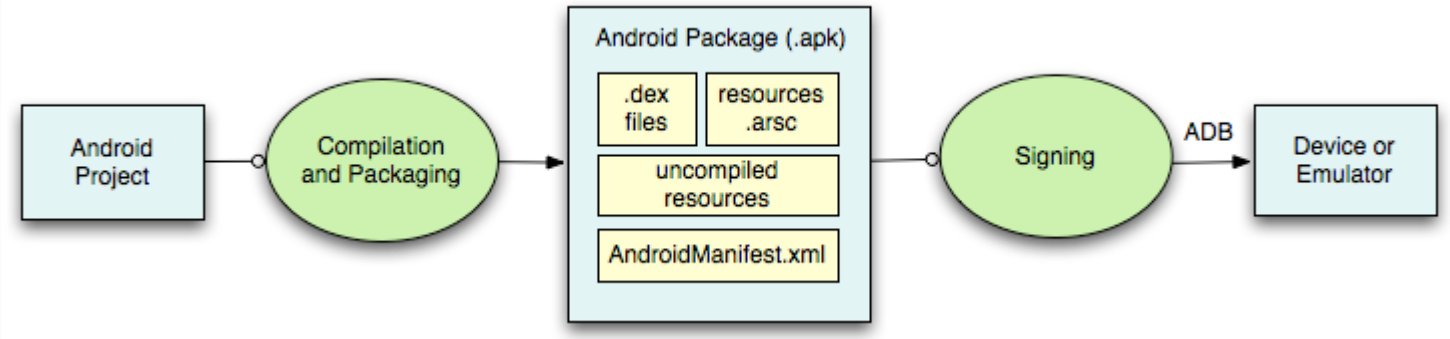
Sicurezza

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

Sicurezza: Compilazione = apk

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

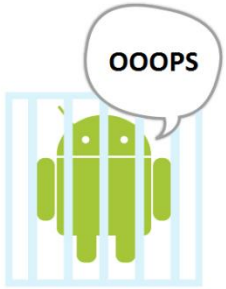


Sicurezza – Decompilare APK

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.



Sicurezza – Decompilare APK

Due approcci

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

- Adozione dei seguenti strumenti

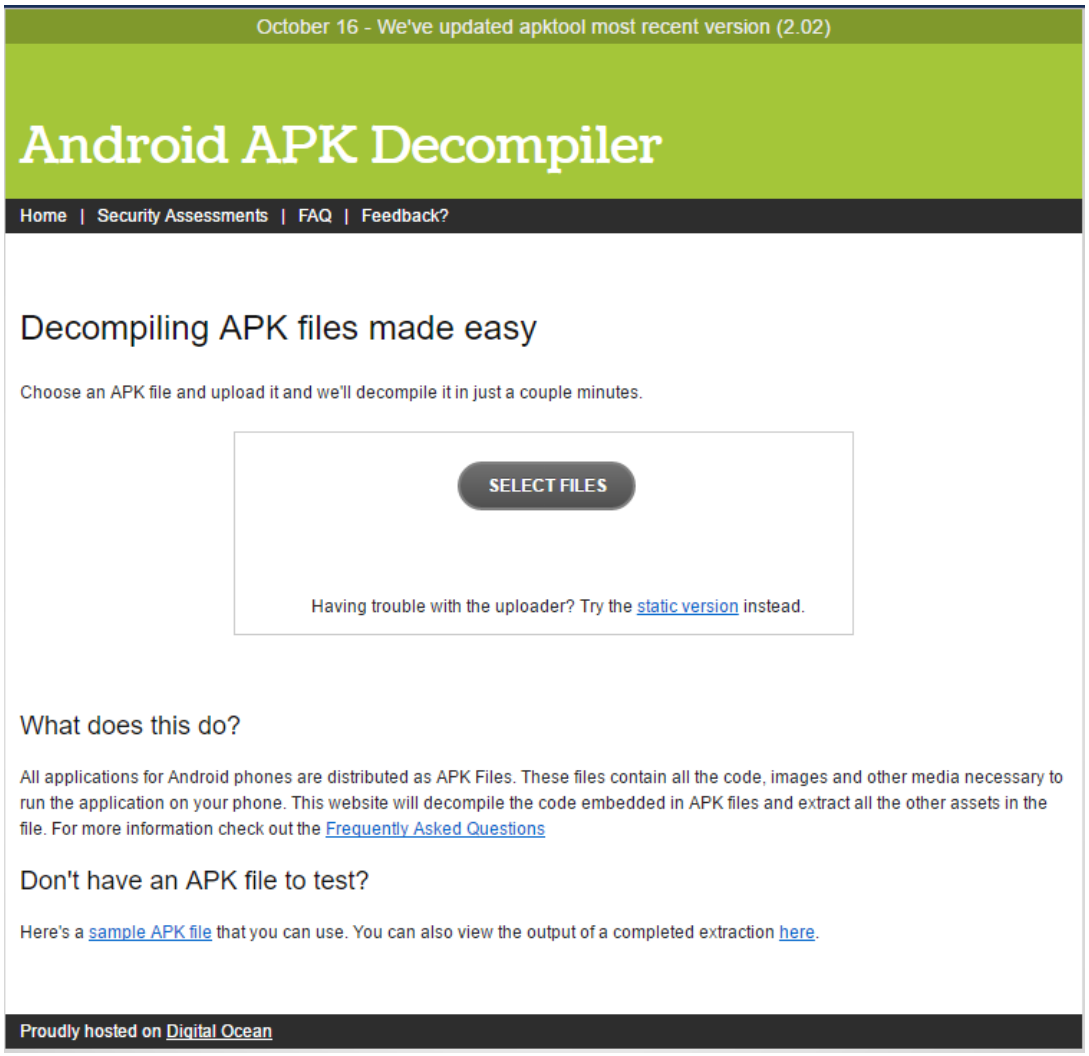
  - apk-tool 
  - dex2jar
  - JD (Java Decompiler) 

  Processo di decompilazione più lungo.

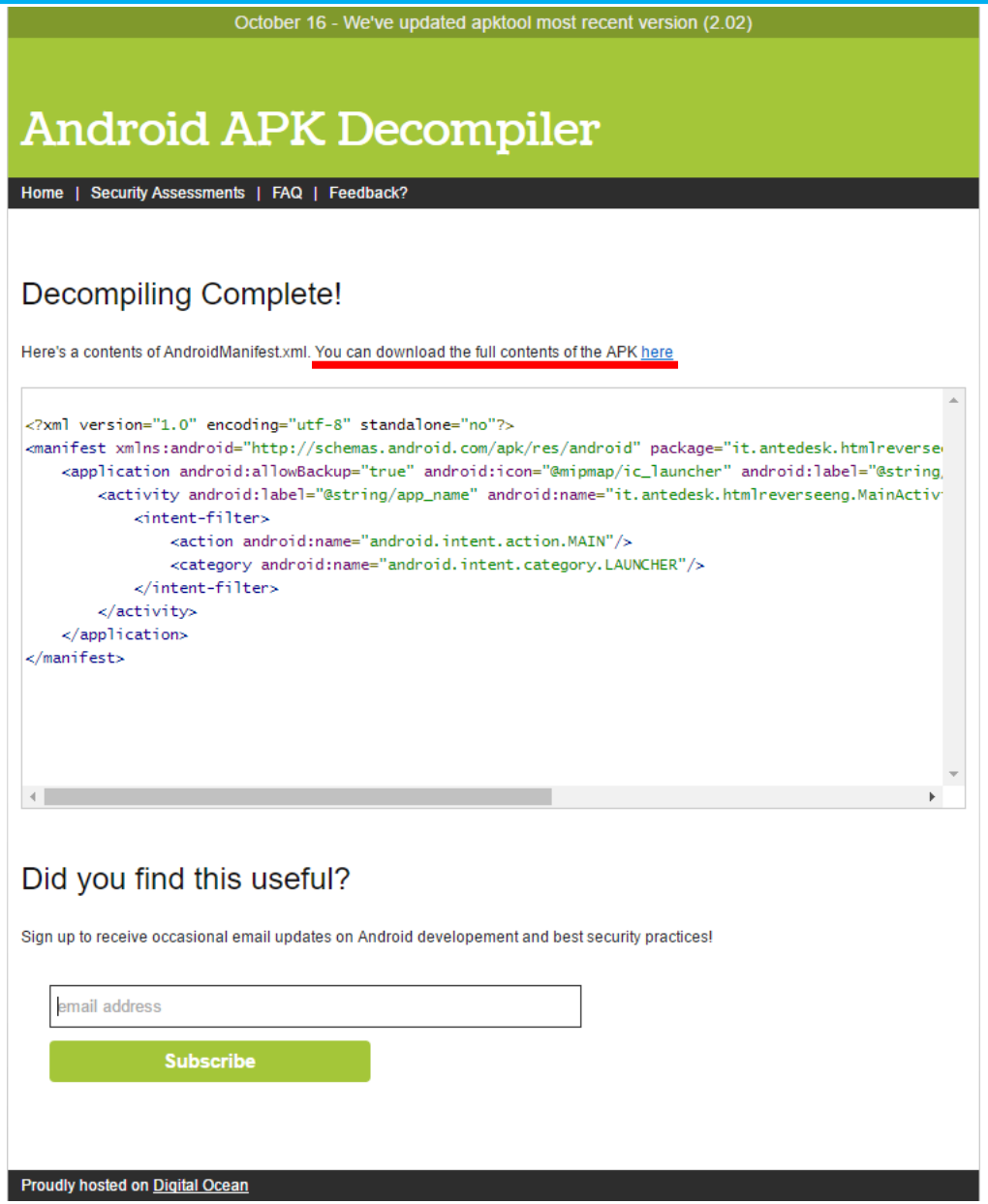
- Referenza: http://www.html.it/articoli/decompilare-ed-offuscare-il-codice-di-unapp-android/

Decompilare APK – Primo metodo

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



Decompilare APK – Primo metodo



Decompilare APK – Seconodo metodo

dex2jar è un insieme di file .bat (sotto Windows) e librerie
Per utilizzarlo è necessario

- modificare l'estensione del 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"

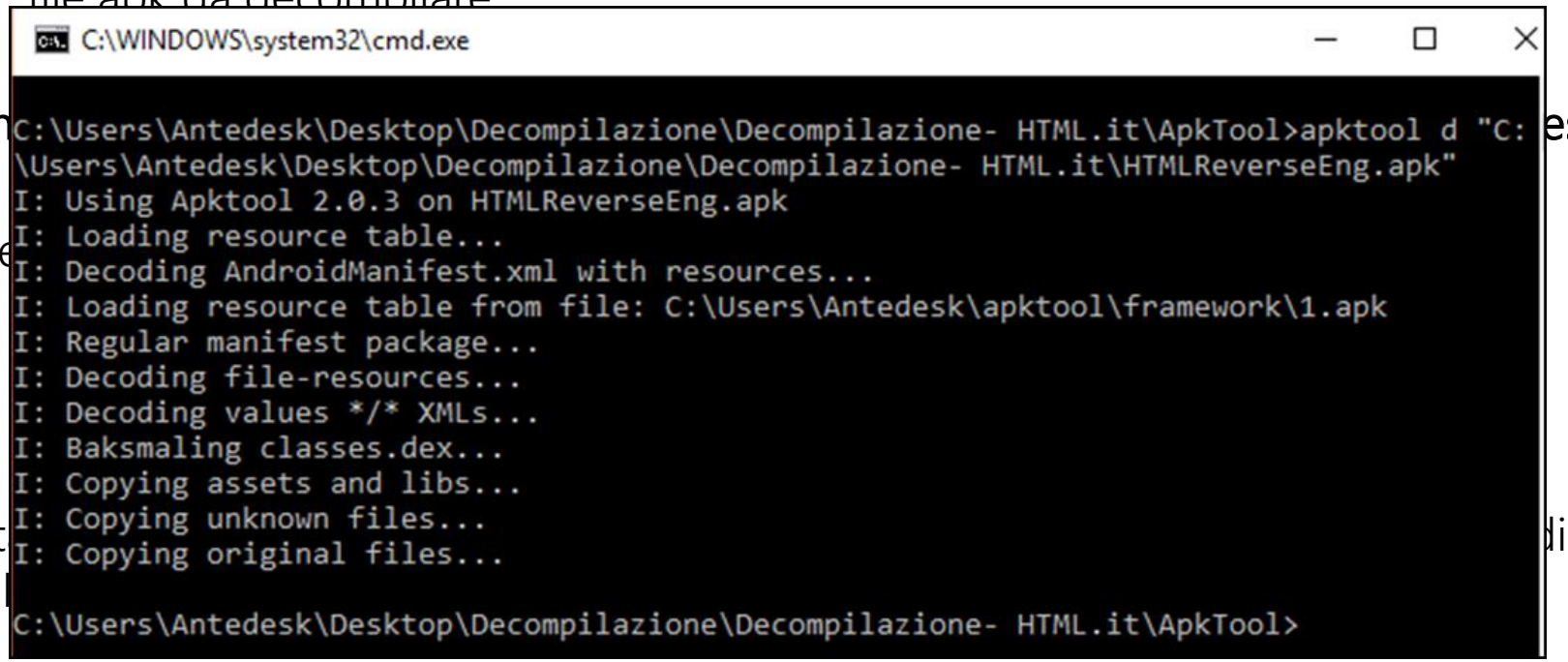
Il risultato sarà la creazione di un file **classes_dex2jar.jar** contenente il codice sorgente dell'app desiderata

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

Decompilare APK – Seconodo metodo

Apktool per funzionare necessita dei seguenti file

- apktool.bat
- apktool_**[Version]**.jar
- file apk da decompilare



Con sh esenti questi

Inserire

Il risult di basso

Risultato:

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

Sicurezza - Problema

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

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

Proguard

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'
    }
}
```

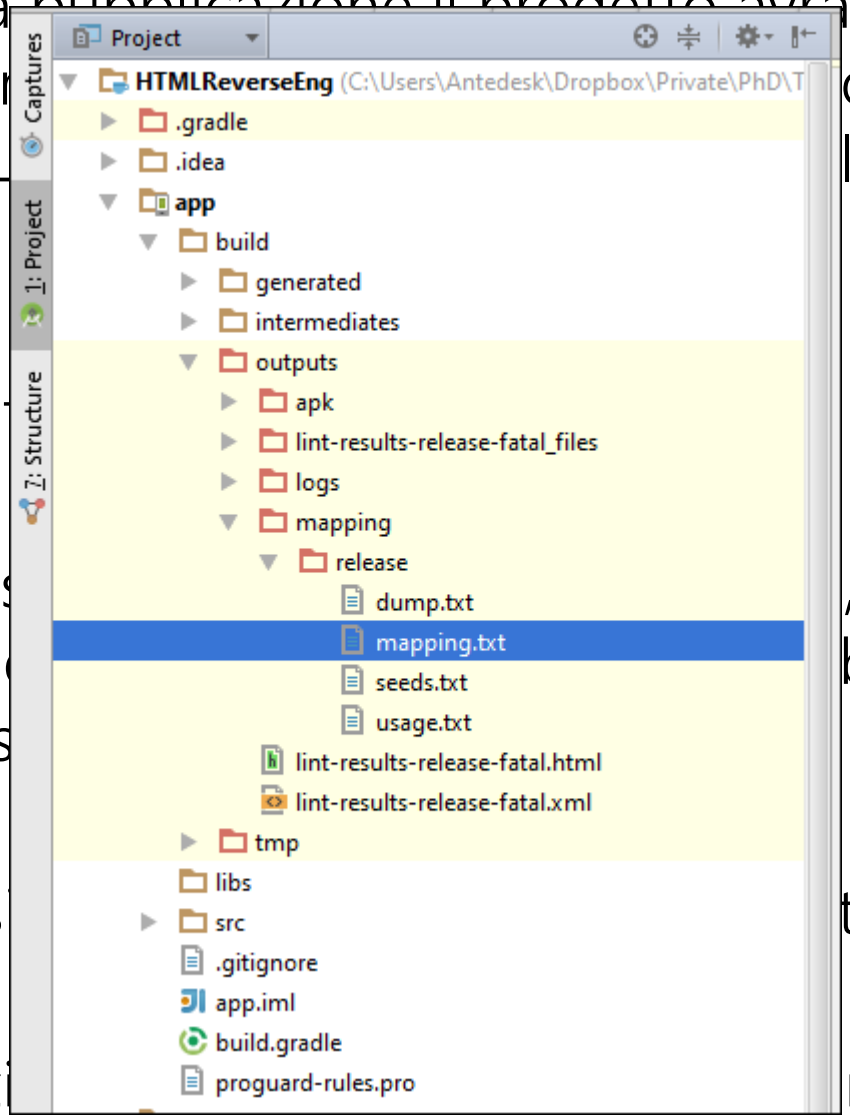
Proguard

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

Creato l'apk per la pubblicazione il progetto avrà una nuova cartella chiamata r o della cartella outputs di build. L le:



- **dump.txt**
  Descrive la stru
- **mapping.txt**
  Lista delle corris , attributi originali e offuscati. Fon bug report per l'app rilasciata s
- **seeds.txt**
  Lista delle class tati offuscati
- **usage.txt**
  Lista delle porzi minati

Proguard - Problematiche

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 eleminare:

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

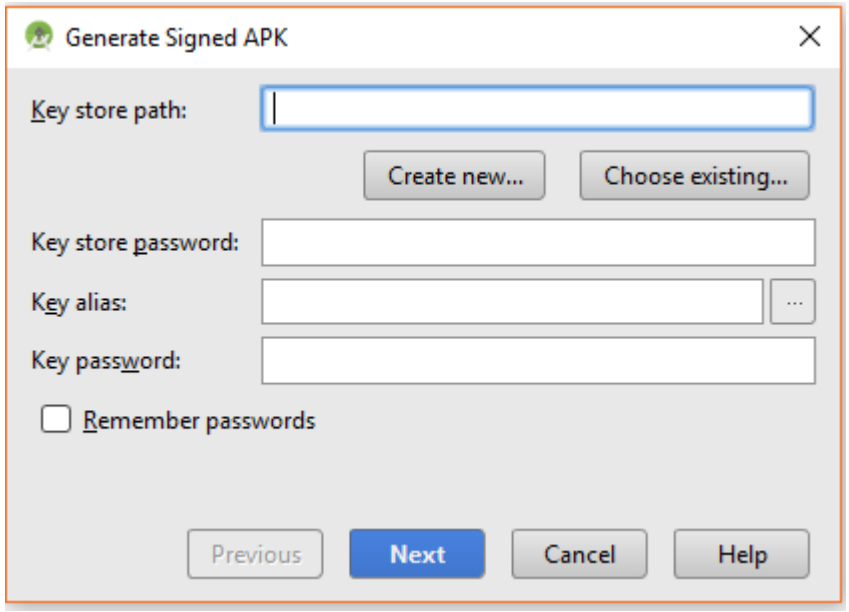
È possibile risolvere il problema aggiungendo al file di configurazione determinate opzioni.

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

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

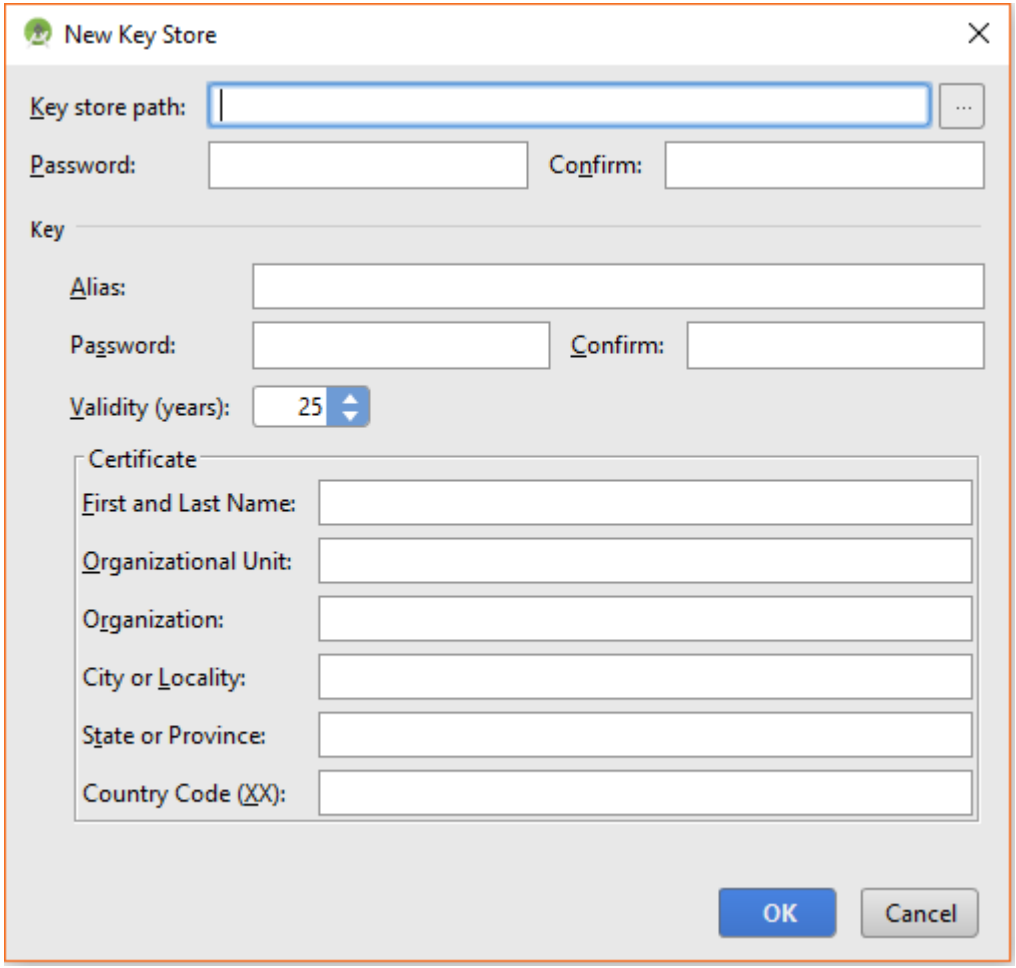
Creazione del file APK

Build -> Generate Signed APK



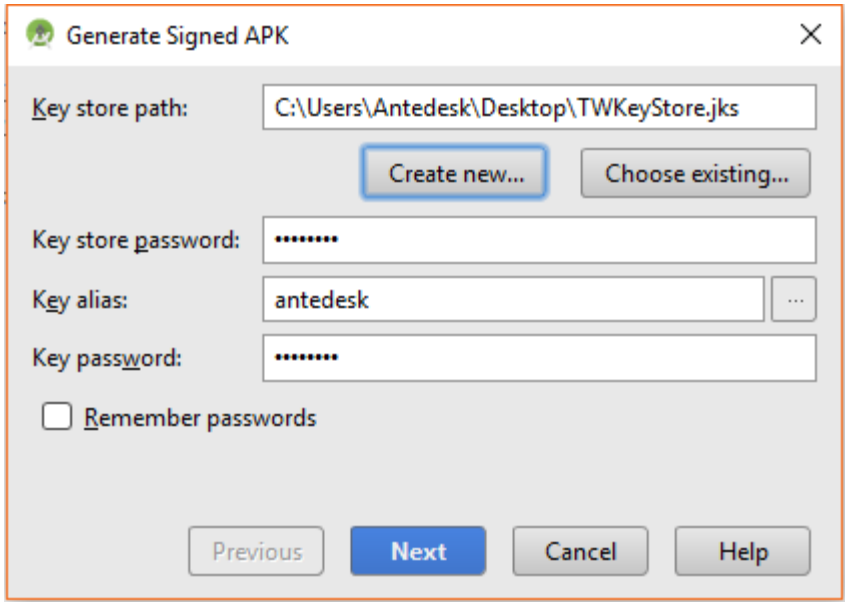
Creazione del file APK

Creazione di una Key Store per firmare l'apk



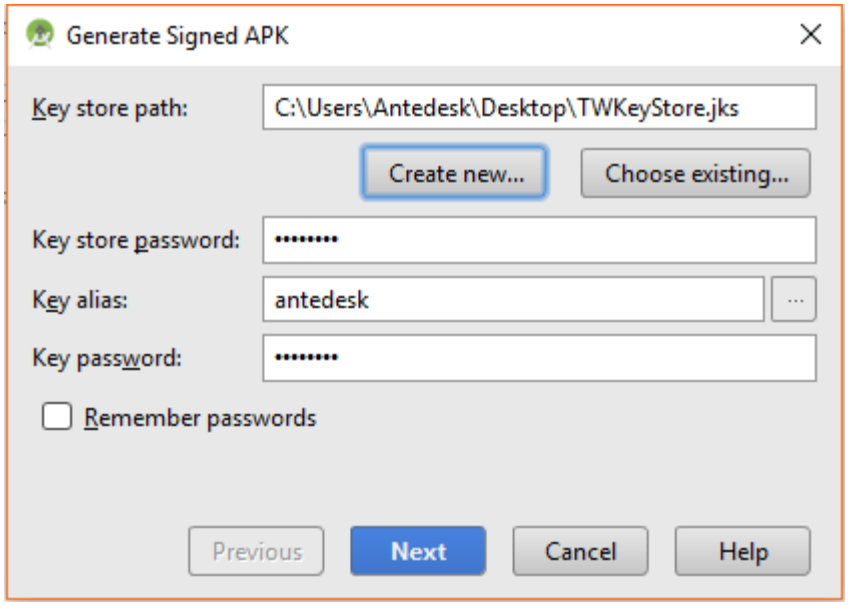
Creazione del file APK

Una volta creata o aggiunta la scheda si autocompilerà



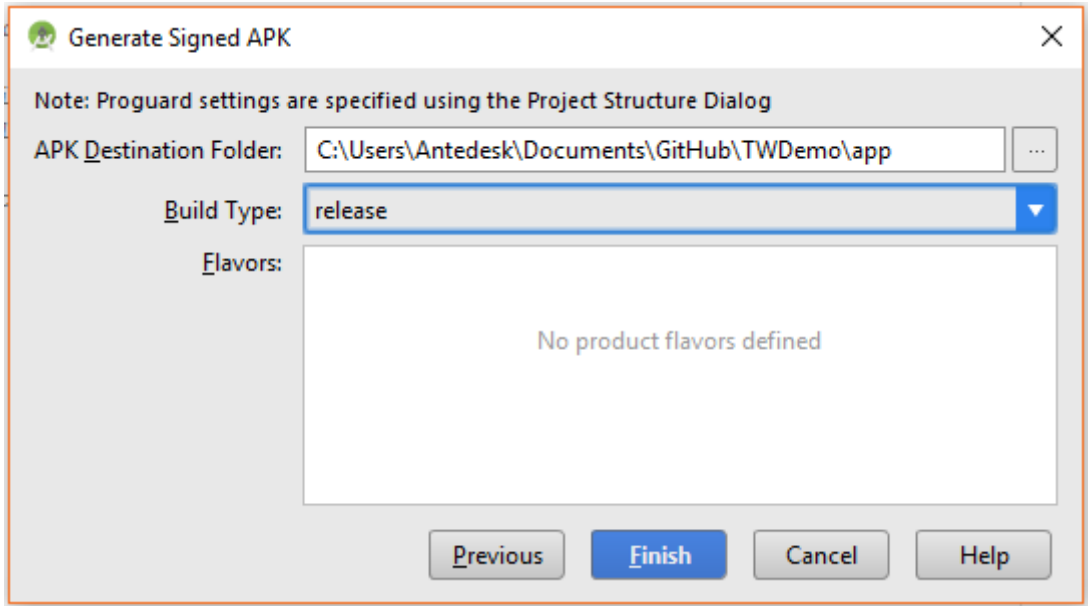
Creazione del file APK

Una volta creata o aggiunta la scheda si autocompilerà



Creazione del file APK

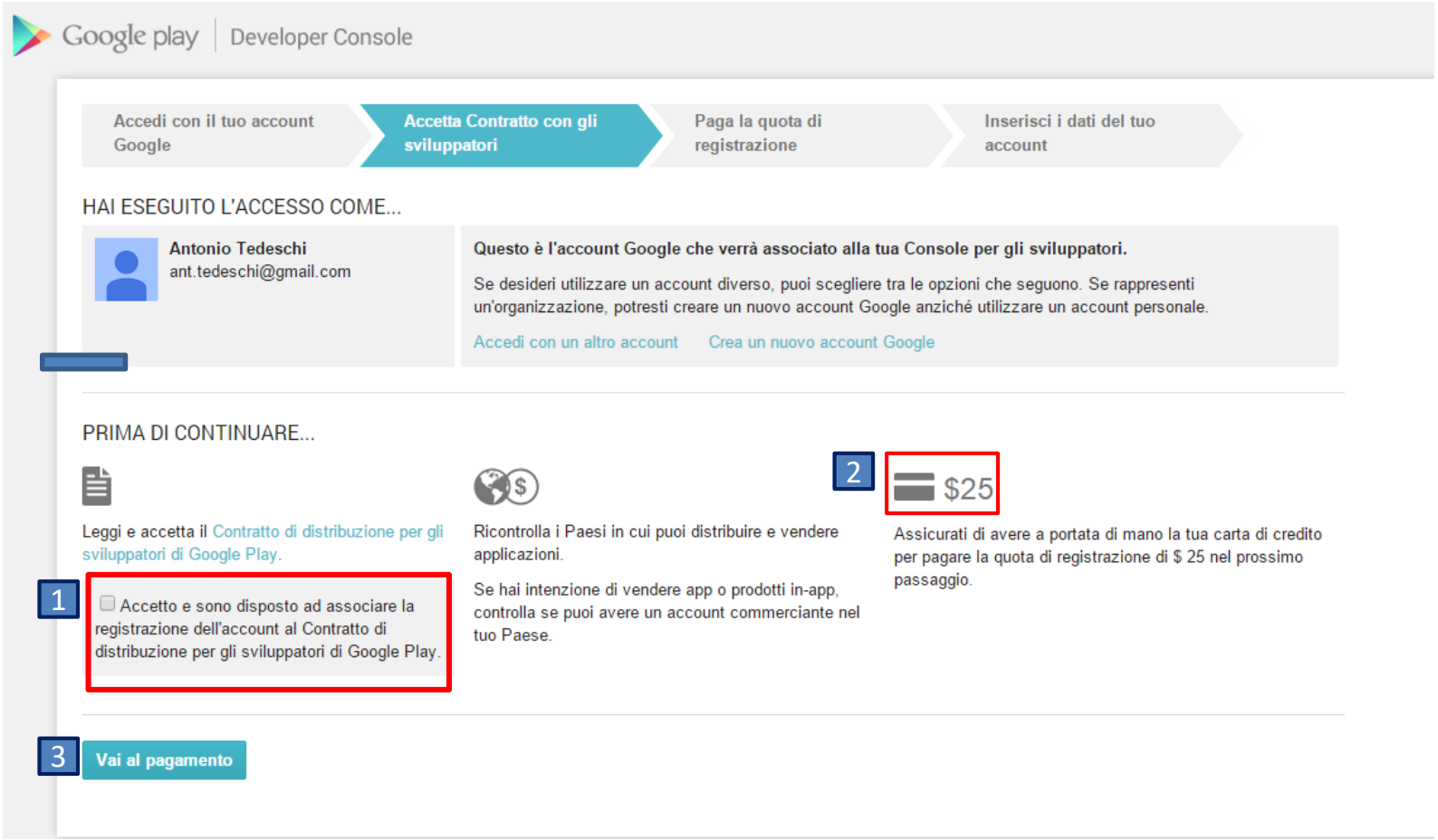
Una volta creata o aggiunta la scheda si autocompilerà



Creazione account developer

- 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

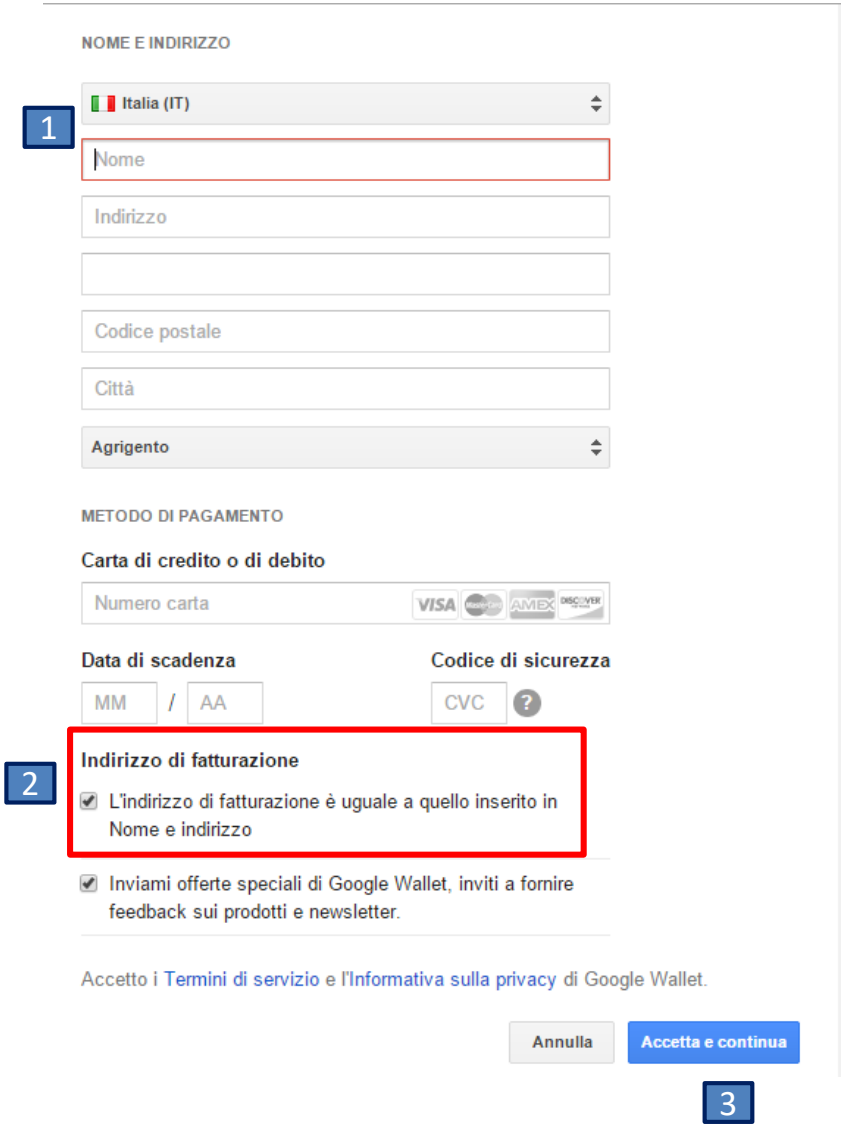


(1) Accettare i termini del contratto

(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

Creazione account developer

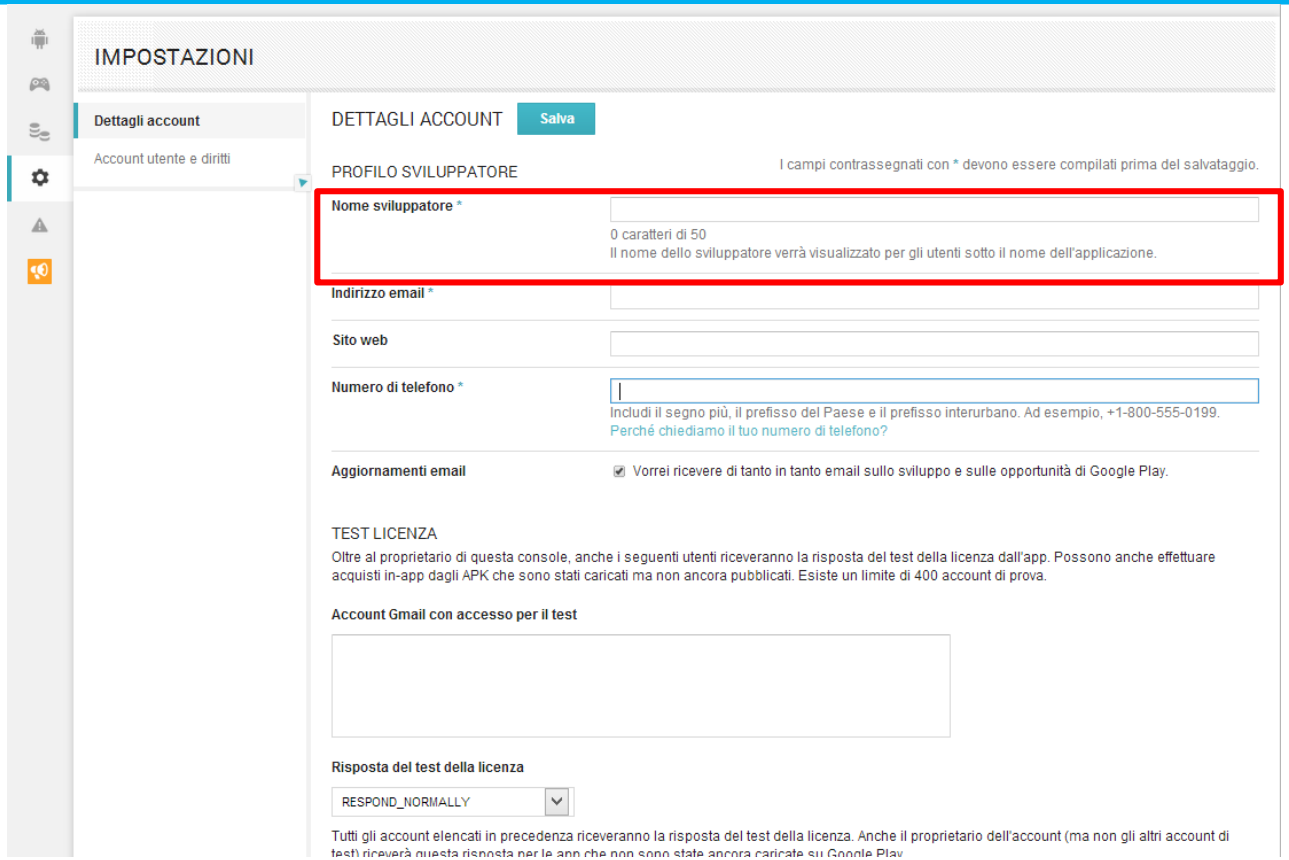


(1) A questo punto impostare il proprio profilo aziendale su Google Wallet, riempendo i campi e inserendo la carta di credito (preferibilmente quella aziendale in quanto sullo stesso profilo di Google Wallet si riceveranno anche i pagamenti dell'applicazione) Informazioni sui metodi di pagamento accettati 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

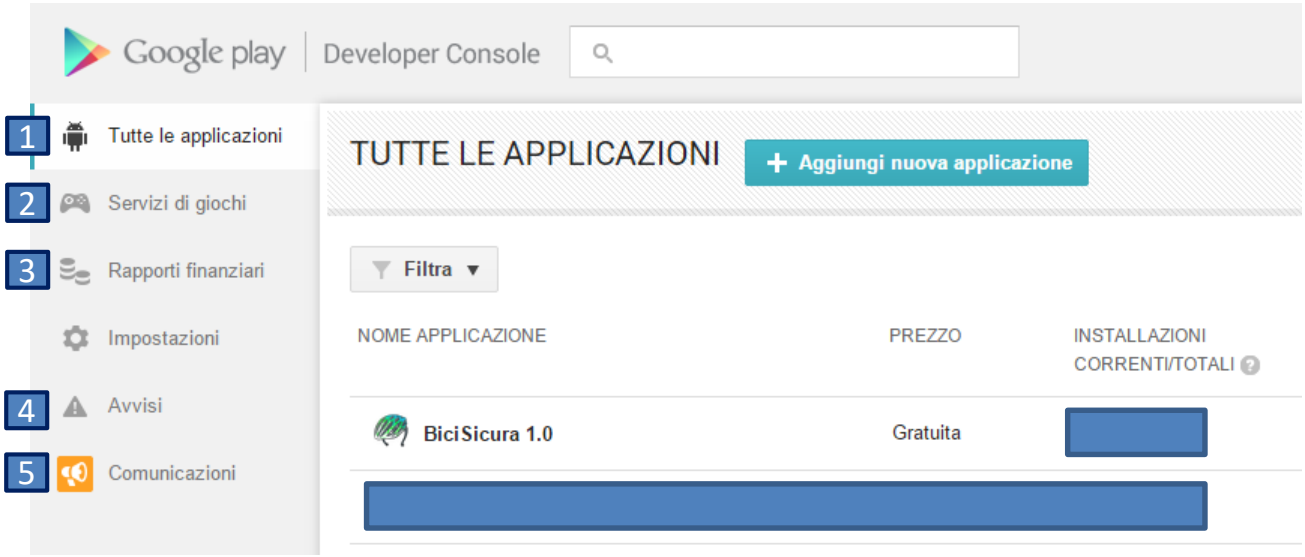
Creazione account developer



- 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 Developer Console



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

Google Developer Console

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.