

---

**UNIVERSITAT POLITÈCNICA DE VALÈNCIA**  
**ESCOLA POLITÈCNICA SUPERIOR D'ALCOI**  
**GRAU EN ENGINYERIA INFORMÀTICA**



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



**Programació dinàmica de port knocking per a dispositius mòbils**

Alumne: Pedro Albuja Rico  
Tutor: José Salvador Blanes Doménech  
Curs acadèmic: 2014 - 2015

---

---

# Índex

|  |           |
|--|-----------|
| <b>1.- Introducció</b>                                   | <b>4</b>  |
| 1.1.- Context  | 4         |
| 1.2.- Objectius  | 5         |
| <b>2.- Port Knocking</b>                                 | <b>6</b>  |
| 2.1.- Què és el port knocking?                           | 6         |
| 2.2.- Servei knockd                                      | 7         |
| 2.3.- Tallafocs per al port knocking                     | 7         |
| <b>3.- Android</b>                                       | <b>9</b>  |
| 3.1.- Què és Android?                                    | 9         |
| 3.2.- Arquitectura                                       | 9         |
| 3.3.- Aplicacions  | 10        |
| 3.4.- Google Play  | 10        |
| 3.5.- Per què elegir Android?                            | 10        |
| 3.6.- Punts negatius                                     | 11        |
| <b>4.- Eines utilitzades</b>                             | <b>13</b> |
| 4.1.- Android Studio                                     | 13        |
| 4.2.- Git  | 14        |
| 4.3.- Gogs   | 14        |
| 4.4.- Java   | 15        |
| 4.5.- Altres eines                                       | 15        |
| <b>5.- Metodologia utilitzada</b>                        | <b>16</b> |
| 5.1.- Model iteratiu e incremental                       | 16        |
| 5.2.- Avantatges   | 16        |
| 5.3.- Desavantatges                                      | 17        |
| <b>6.- Planificació</b>                                  | <b>18</b> |
| 6.1.- Planificació per objectius                         | 18        |
| 6.2.- Diagrama de Gantt                                  | 18        |
| <b>7.- Estructura i elements d'un projecte d'Android</b> | <b>20</b> |
| 7.1.- Estructura d'un projecte d'Android                 | 20        |
| 7.2.- Elements d'una aplicació d'Android                 | 21        |
| <b>8.- Desenvolupament de l'aplicació</b>                | <b>23</b> |
| 8.1.- Anàlisi de requisits                               | 23        |
| 8.1.1.- Requisits de l'aplicació                         | 23        |
| 8.1.2.- Casos d'ús                                       | 24        |
| 8.2.- Disseny  | 26        |
| 8.2.1.- Arquitectura de l'aplicació                      | 26        |

---

|  |           |
|--|-----------|
| 8.2.2.- Diagrama de fluxe                    | 28        |
| 8.3.- Implementació                          | 29        |
| 8.3.1.- Emmagatzemament de dades             | 29        |
| 8.3.2.- Implementació de la funcionalitat    | 30        |
| 8.3.2.- Implementació gràfica de l'aplicació | 31        |
| 8.4.- Validació i test                       | 33        |
| 8.5.- Problemes trobats                      | 33        |
| <b>9.- Manuals de l'aplicació</b>            | <b>35</b> |
| 9.1.- Manual d'instal·lació                  | 35        |
| 9.2.- Manual d'ús                            | 36        |
| <b>10.- Publicació a Google Play</b>         | <b>39</b> |
| <b>11.- Open Source</b>                      | <b>40</b> |
| 11.1.- Llicència                             | 40        |
| 11.2.- Publicació a GitHub                   | 41        |
| <b>12.- Possibles ampliacions</b>            | <b>43</b> |
| 12.1.- Ampliacions en procés                 | 43        |
| 12.2.- Ampliacions futures                   | 43        |
| <b>13.- Conclusions</b>                      | <b>44</b> |
| <b>14.- Bibliografia</b>                     | <b>45</b> |

# 1.- Introducció

## 1.1.- Context

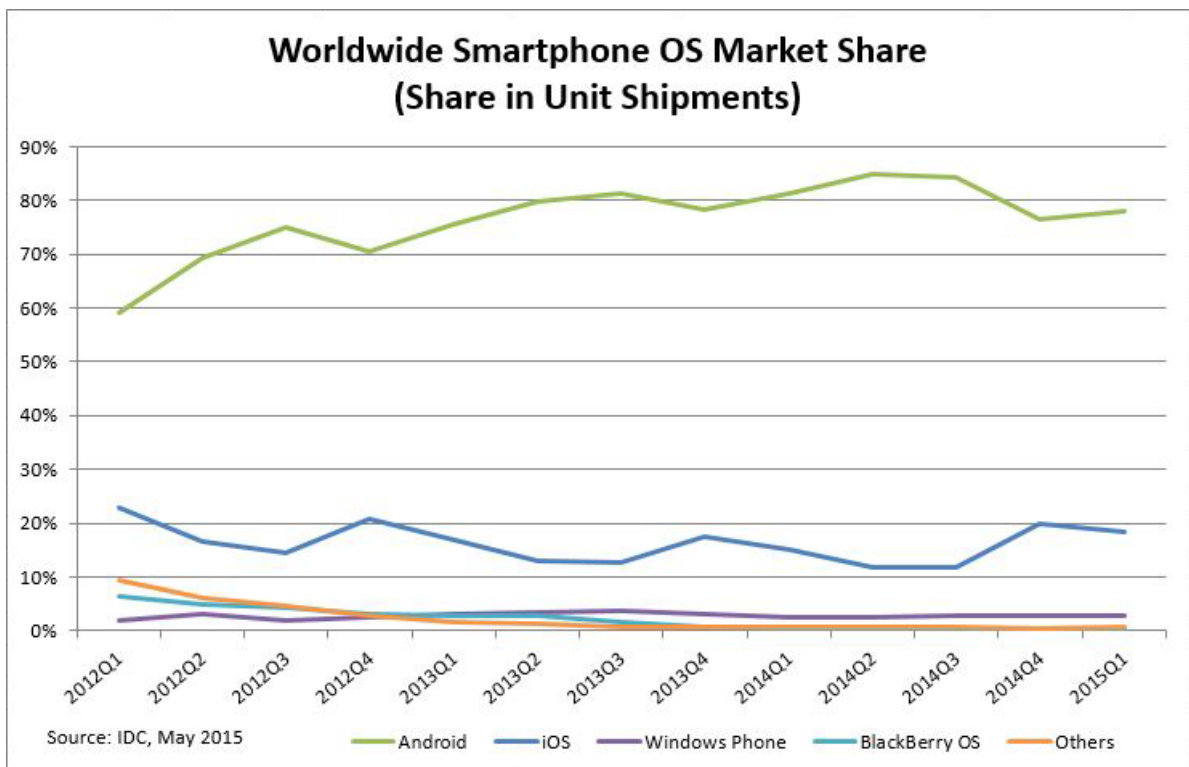
En l'actualitat els smartphones estan convertint-se en l'eina tecnològica més utilitzada. Amb ells podem fer pràcticament qualsevol cosa per a la qual abans utilitzàvem un ordinador, com per exemple per a consultar el correu, llegir notícies a periòdics digitals, xatejar, etc.

Aquest tipus de dispositius s'han convertit pràcticament en un objecte imprescindible amb el qual anem a tots els llocs.

Aquest últim punt és el que fa que siga realment interessant utilitzar els smartphones com a clients de port knocking, ja que podem modificar les regles del tallafocs del nostre ordinador o servidor des de qualsevol lloc sense la necessitat d'un ordinador.

Altre dels punts a tindre en conte és la varietat de sistemes operatius per a dispositius mòbils que existeixen, com per exemple Android, iOS, Windows Phone, BlackBerry OS, etc.

En el següent gràfic es pot veure quina és la quota de mercat dels diferents sistemes operatius mòbils i com ha anat evolucionant en els últims anys.



Com podem observar al gràfic proporcionat per l'IDC (International Data Corporation) cal destacar Android ja que és el que més quota té amb un 78%, seguit iOS amb un 18,3%, aquest motiu entre altres que seran analitzats més avant i seran els que ens faran decidir-nos per aquest sistema operatiu per a implementar l'aplicació per al client de port knocking.

---

## 1.2.- Objectius

Amb la realització d'aquest treball es volen aconseguir diversos objectius, els quals seran descrits a continuació:

- Ja que a aquest treball va a realitzar-se una aplicació per a Android aquest primer objectiu consisteix en aprendre el funcionament intern d'aquest sistema operatiu.
- Aprendre el funcionament intern de les aplicacions així com les eines per a poder crear-les.
- Dissenyar gràficament una aplicació agradable per l'usuari segons els estàndards de disseny de Material Design.
- Aprendre a distribuir aplicacions mitjançant Google Play.
- Aprendre a utilitzar eines de control de versions com per exemple git.
- Altre dels objectius és aprofundir amb els coneixements relacionats amb el port knocking, és a dir, l'ús de ports, regles del tallafocs i configuració del servei de port knocking.

---

## 2.- Port Knocking

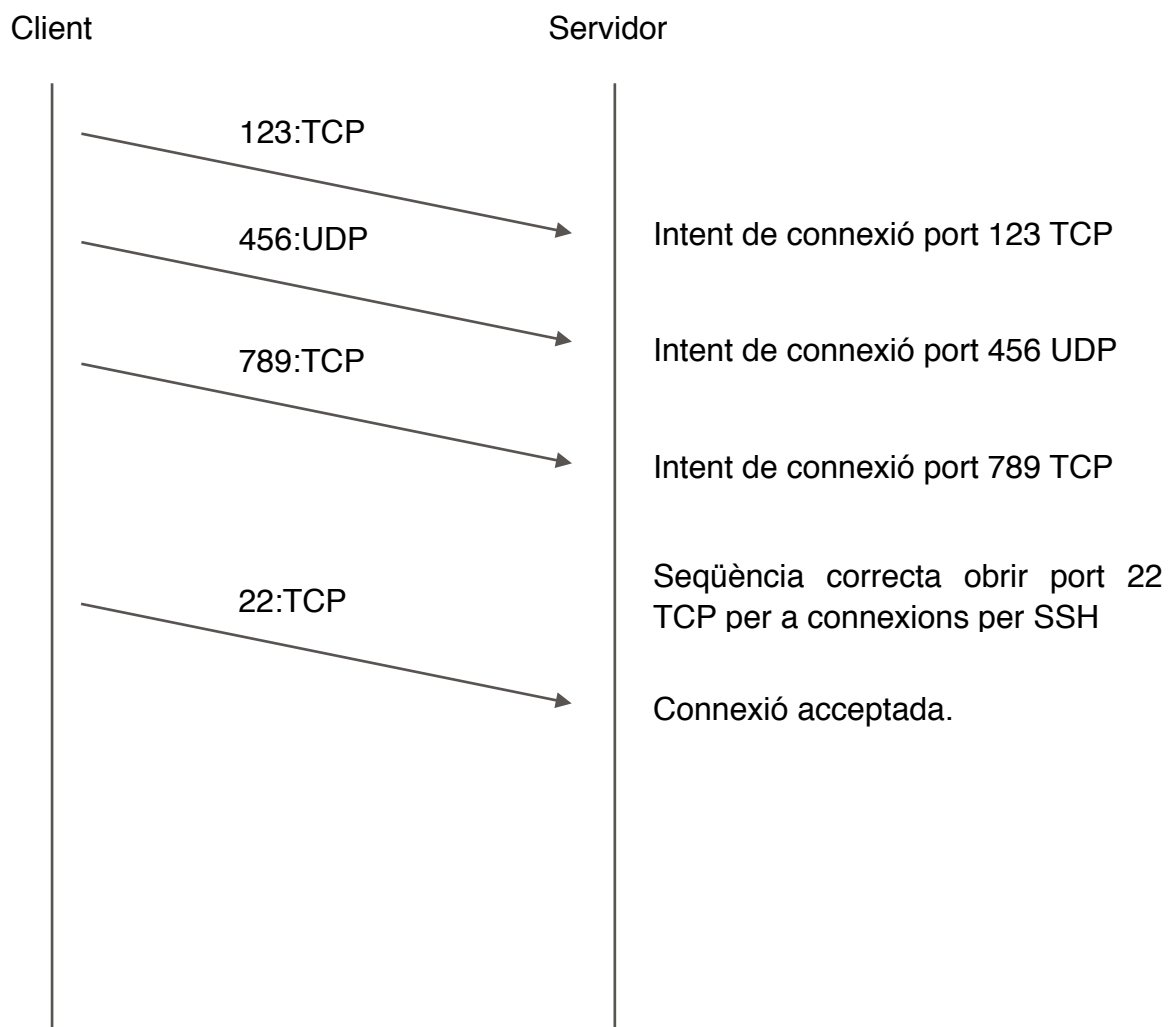
### 2.1.- Què és el port knocking?

El port knocking o colpejament de ports és un mecanisme per a obrir ports externament en un tallafocs mitjançant una seqüència preestablerta d'intents de connexió a uns ports que estan tancats. Una vegada l'ordinador o servidor rep una seqüència correcta el que fa és modificar les regles del tallafocs per a permetre la connexió al port establert.

El propòsit principal del port knocking és previndre que un possible atacant pugui escanejar els ports en busca d'un servei vulnerable, utilitzant el port knocking l'atacant no trobarà cap port obert que pugui ser vulnerable.

El port knocking s'implementa configurant un servei perquè revisei el log del tallafocs per detectar aquesta seqüència d'intents de connexió i obrir el port configurat.

A continuació es mostra un esquema del funcionament del port knocking on s'accepta una connexió al port 22 després de enviar la seqüència correcta.



---

## 2.2.- Servei knockd

Com ja hem nomenat abans per a implementar el port knocking hem de configurar un servei per a revisar el log del tallafocs. En aquest cas utilitzarem el knockd, aquest servei es pot instal·lar a qualsevol màquina que utilitzi linux mitjançant el gestor de paquets que tinguem al nostre sistema operatiu. En aquest cas utilitzaré Linux Mint i el gestor APT.

Per a poder instal·lar aquest servei utilitzarem el següent comandament com a root.

```
apt-get install knockd
```

Una vegada instal·lat el paquet editarem l'arxiu de configuració següent com a root "/etc/default/knockd" i a la opció START\_KNOCKD li canviarem el valor 0 per 1.

Seguidament configurarem les seqüències de port knocking al següent arxiu: "/etc/knockd.conf"

```
[openSSH]
sequence = 123:tcp,456:udp,789:tcp
seq_timeout = 5
command = /sbin/iptables -I INPUT -p tcp --dport 22 -j ACCEPT -m comment --comment "Regla Port Knocking SSH"
tcpflags = syn
cmd_timeout = 120
stop_command = /sbin/iptables -D INPUT -p tcp --dport 22 -j ACCEPT -m comment --comment "Regla Port Knocking SSH"

[openFTP]
sequence = 123:udp,456:tcp,789:udp
seq_timeout = 5
command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 21 -j ACCEPT -m comment --comment "Regla Port Knocking FTP"
tcpflags = syn
cmd_timeout = 120
stop_command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 21 -j ACCEPT -m comment --comment "Regla Port Knocking FTP"
```

En aquest arxiu hi han dos regles configurades per als ports 22 i 21 (SSH i FTP), amb les seqüències següents:

SSH: 123:tcp, 456:udp, 789:tcp

FTP: 123:udp, 456:tcp, 789:udp

En els dos casos es tancaran els ports passats 120 segons. En quant al SSH es permetrà l'accés a tothom mentre que el FTP sols a qui haja enviat la seqüència.

Finalment posarem el servei en funcionament executant el següent comandament com a root:

```
service knockd start
```

## 2.3.- Tallafocs per al port knocking

Una vegada ja configurat el knockd a la part que actua com a servidor ja sols ens queda configurar el tallafocs. El que farem serà establir les polítiques per defecte en DROP i afegir una regla que manté totes les connexions establertes ja que en la configuració del knockd li hem dit que tanque els ports passats 120 segons.

---

En el meu cas el que faré serà executar el següent script al iniciar el servidor de forma que establisca les regles del tallafocs.

```
#!/bin/bash
/sbin/iptables -F

/sbin/iptables -P INPUT DROP
/sbin/iptables -P FORWARD DROP
/sbin/iptables -P OUTPUT ACCEPT

/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```



---

## 3.- Android

### 3.1.- Què és Android?

Android és un sistema operatiu basat en el nucli Linux. Va ser dissenyat principalment per a dispositius mòbils amb pantalla tàctil, com smartphones o tablets, actualment també s'utilitza a rellotges, televisors i automòbils. Inicialment va ser desenvolupat per Android Inc, empresa que Google va recolzar econòmicament i més tard va comprar.

### 3.2.- Arquitectura

Els principals components del sistema operatiu Android són els següents:

- **Nucli Linux:** Android depèn de Linux per als serveis base del sistema com seguretat, gestió de memòria, gestió de processos, pila de xarxa i model de controladors. El nucli també actua com una capa d'abstracció entre el hardware i la resta de la pila de software.
- **Aplicacions:** les aplicacions base inclouen un client de correu electrònic, programa de SMS, calendari, mapes, navegador, contactes i altres. Totes les aplicacions estan escrites en llenguatge de programació Java.
- **Marc de treball d'aplicacions:** els programadors tenen accés complet als mateixos APIs del framework que utilitzen les aplicacions base. L'arquitectura està dissenyada per simplificar la reutilització de components; qualsevol aplicació pot publicar les seues capacitats i qualsevol altra aplicació pot després fer ús d'aquestes capacitats (subjecte a regles de seguretat del framework). Aquest mateix mecanisme permet que els components siguin reemplaçats per l'usuari.
- **Llibreries:** Android inclou un conjunt de llibreries de C / C ++ utilitzades per diversos components del sistema. Aquestes característiques s'exposen als programadors a través del marc de treball d'aplicacions d'Android. Algunes d'aquestes llibreries són: System C library (implementació llibreria C estàndard), llibreries de gràfics, 3D i SQLite, entre altres.
- **Runtime d'Android:** Android inclou un set de llibreries base que proporcionen la major part de les funcions disponibles a les llibreries base del llenguatge Java. Cada aplicació Android corre el seu propi procés, amb la seva pròpia instància de la màquina virtual Dalvik. Dalvik s'ha escrit de manera que un dispositiu pot executar múltiples màquines virtuals de forma eficient. Dalvik executa arxius en el format Dalvik Executable (.dex), el qual està optimitzat per memòria mínima. La Màquina Virtual està basada en registres i executa classes compilades pel compilador de Java que han estat transformades al format .dex. Actualment des de l'última versió d'Android anomenada Lollipop aquesta màquina virtual ha sigut reemplaçada per una nova anomenada ART.

---

### 3.3.- Aplicacions

Les aplicacions per a Android estan programades principalment utilitzant el llenguatge de programació Java amb Android Software Development Kit (Android SDK).

Per a la programació d'aplicacions per a Android existeixen principament dos alternatives. La primera és Eclipse amb el plugin Android Developer Tools (ADT) la qual ja esta en desús ja que per part de Google han deixat de donar suport al plugin ADT i van decidir crear el seu propi IDE anomenat Android Studio. Aquest propi IDE és la segona de les alternatives que tenim. Per a realitzar aquest treball es farà ús d'Android Studio ja que té suport oficial per part de Google.

### 3.4.- Google Play

Google Play és una plataforma de distribució digital d'aplicacions mòbils per a dispositius amb sistema operatiu Android. Aquesta plataforma permet al programadors distribuir les seues aplicacions d'una manera simple, ràpida i segura. Per a poder distribuir aplicacions a Google Play sols fa falta registrar-se, pagar una xicoteta i única quota de 25\$ i finalment pujar la nostra aplicació. Una vegada ja ens em registrat podem pujar tantes aplicacions com vullguem amb el mateix conte.

### 3.5.- Per què elegir Android?

El principals motius per els quals s'ha elegit aquest sistema operatiu per realitzar el treball són els següents:

- En primer lloc com ja hem comentat a la introducció Android és el sistema operatiu que més quota de mercat té i per tant la nostra aplicació serà accessible per a molta més gent.
- El segon punt a valorar és el tema de la distribució d'aplicacions en el cas de Android tenim Google Play, en cas d'haver optat per el segon candidat de la llista de quota de mercat (iOS) tendríem l'App Store. Per a poder publicar aplicacions a Google Play sols hem de registrar-nos i pagar una xicoteta i única quota de 25\$, en el cas de l'App Store la quota seria de 99\$ anuals. Com es pot veure és molt més barat el cas de Google Play. En el cas de voler vendre la nostra aplicació si que hi hauria que tindre el conte l'App Store ja que genera més ingressos però en el nostre cas serà una aplicació gratuïta.
- Altre dels punts a tindre en conte encara que en el meu cas no és un problema però per a altres programadors podria ser-ho és que per a programar aplicacions per a iOS és necessari tindre un ordinador de Apple ja que sinó no serà possible crear aplicacions per a aquest sistema operatiu. En el cas de Android podem programar utilitzant Linux, Windows i Mac OS ja que Android Studio està disponible per als tres sistemes operatius.

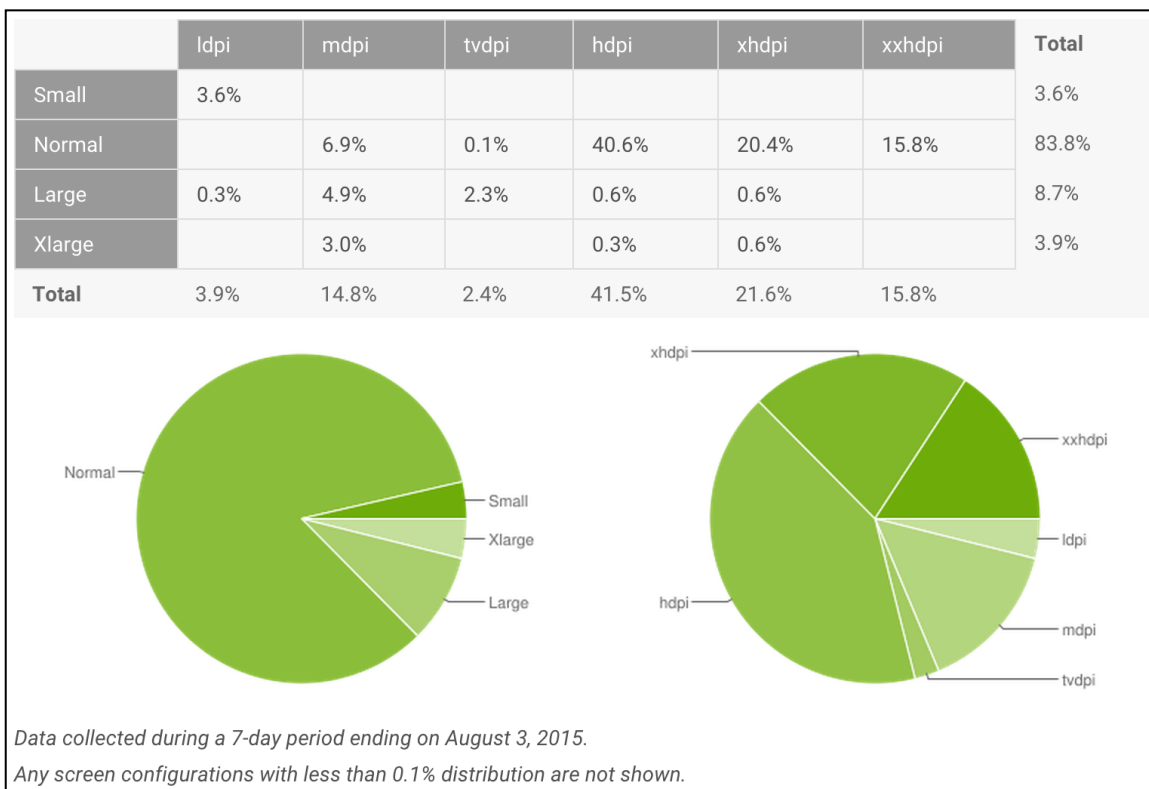
- Altre dels punts que m'ha fet decidir-me per utilitzar Android és que els dispositius mòbils que jo tinc utilitzen aquest sistema operatiu. Açò podria tindre una solució ja sent usuari de Mac OS les eines de programació per iOS al igual que les d'Android proporcionen un emulador, però òbviament és molt millor poder provar la nostra aplicació en un dispositiu físic.
- L'últim punt són els llenguatges de programació en el cas d'Android s'utilitza Java un llenguatge el qual he après durant la realització del grau i el qual és molt familiar per a mi. En el cas d' iOS el llenguatge utilitzat és Swift.

### 3.6.- Punts negatius

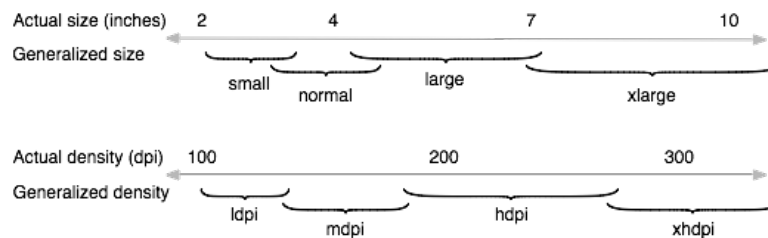
Finalment anem a tractar els punts negatius d'utilitzar Android a l'hora de programar aplicacions per a aquest sistema operatiu. El punts que anem a tractar són els següents:

- Un dels problemes que ens trobem a Android i que no ocorren a altres sistemes operatius com iOS és la quantitat de dispositius diferents que existeixen. En el cas d' iOS sols tenim els dispositius fabricats per Apple. Però en el cas tenim multitud de dispositius fabricats per diferents fabricants com per exemple Samsung, LG, Sony, HTC... El problema que ens genera tindre tanta quantitat de dispositius és que no tots són iguals, ja siga pel tamany de pantalla, la resolució d'aquesta o pels distints tipus de sensors que tinguem disponibles als diferents dispositius.

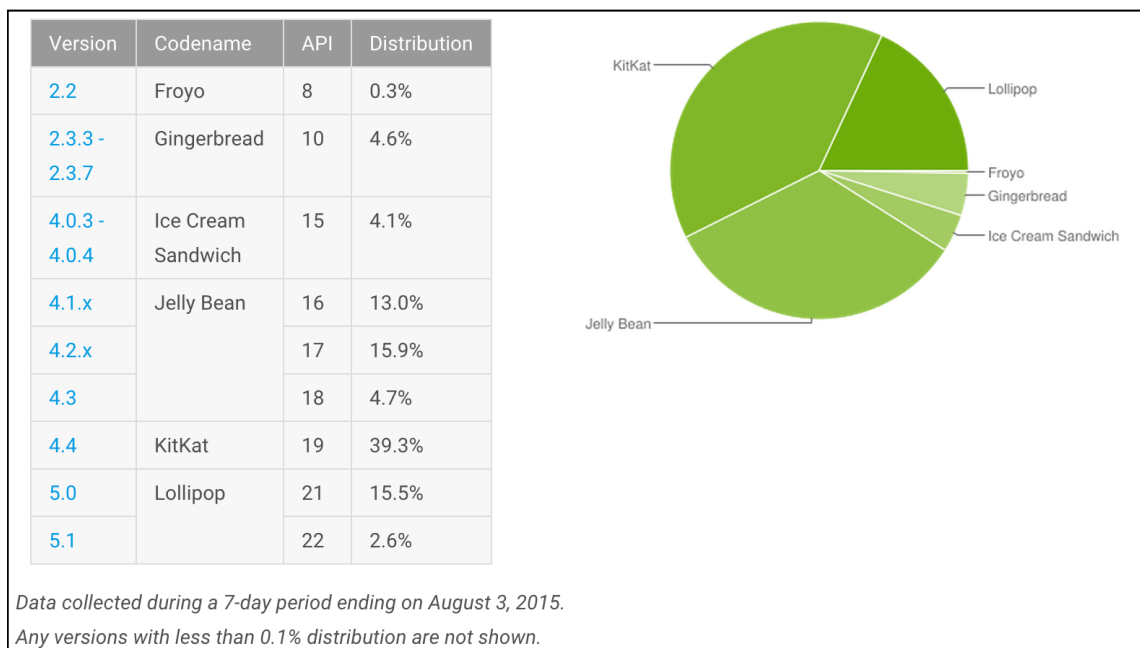
A continuació es mostra un gràfic proporcionat per a la web d'Android en el qual es pot observar la relació entre el tamany de pantalla i la densitat de píxels.



Per a poder entendre l'anterior gràfica adjunte les equivalències dels tamanys de pantalla amb el seu equivalent en polzades i en dpi.



- Altre dels punts negatius és la fragmentació entre versions d'Android, aquesta fragmentació es produïx ja que les actualitzacions del dispositius depenen dels fabricants i les companyies telefòniques i en la majoria dels casos deixen d'actualitzar els dispositius molt prompte. A continuació observarem altre gràfic proporcionat per la web d'Android on es pot observar el percentatge de dispositius segons la versió d'Android que utilitzen.



Aquest gràfic el tindrem en conte més avant per a elegir l'API que utilitzarem.

## 4.- Eines utilitzades

### 4.1.- Android Studio

Android Studio és l'IDE (Integrated Development Environment) que utilitzarem per a programar l'aplicació. Aquest IDE és l'oficial i recomanat per a programar aplicacions per a Android. Està disponible per als diferents sistemes operatius del mercat Linux, Mac OS i Windows i es pot descarregar des de la següent pàgina web: <https://developer.android.com/sdk/index.html>

Aquest software està basat en el software IntelliJ IDEA de JetBrains.

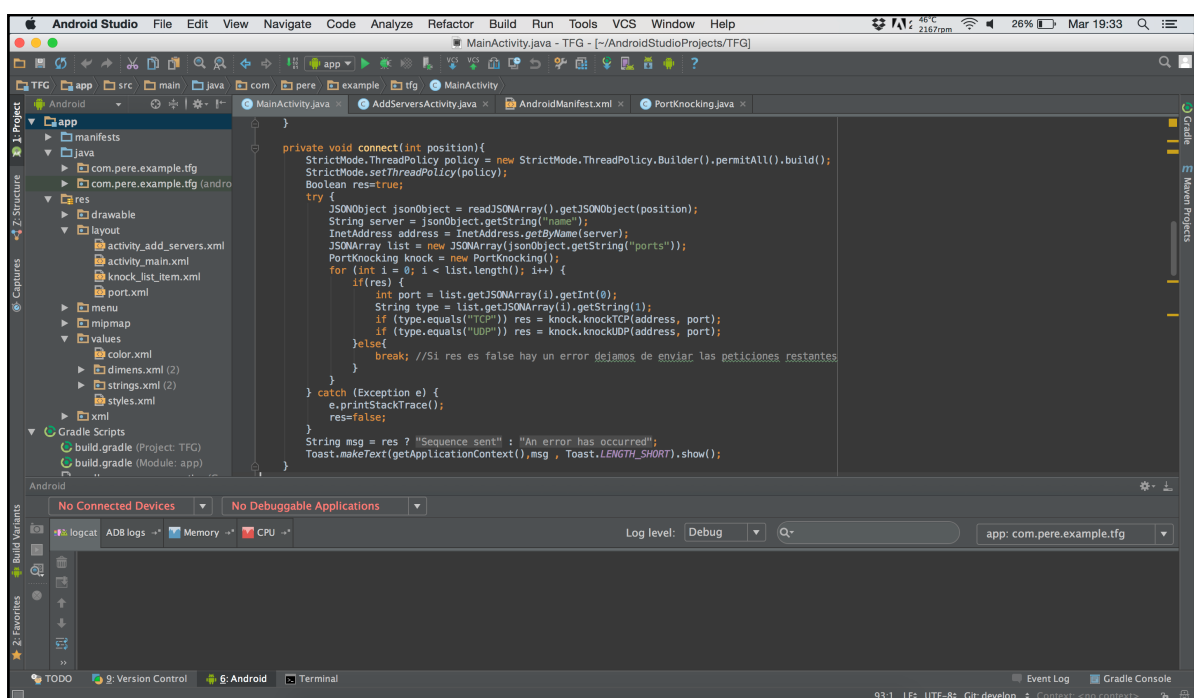
El requisits mínims per a instal·lar-lo són els següents:

- 2 GB de RAM (4 GB recomanats).
- 400 MB d'espai en disc.
- 1 GB para Android SDK.
- Monitor de 1280x800.
- Java Development Kit 7.

A més per a Linux és necessària la següent llibreria:

- [GNU Library C 2.15](#) o superior

Aquest IDE ens proporciona multitud d'eines per a programar per a Android, ens proporciona un gestor per a descarregar el SDK, tenim un gestor de màquines virtuals on podem crear qualsevol tipus de dispositiu amb les característiques que vullguem com per exemple la resolució de pantalla o la versió que executen de forma que es poden crear diferents dispositius virtuals per a poder comprovar com es voria la nostra aplicació al executar-la en un dispositiu amb eixes característiques.



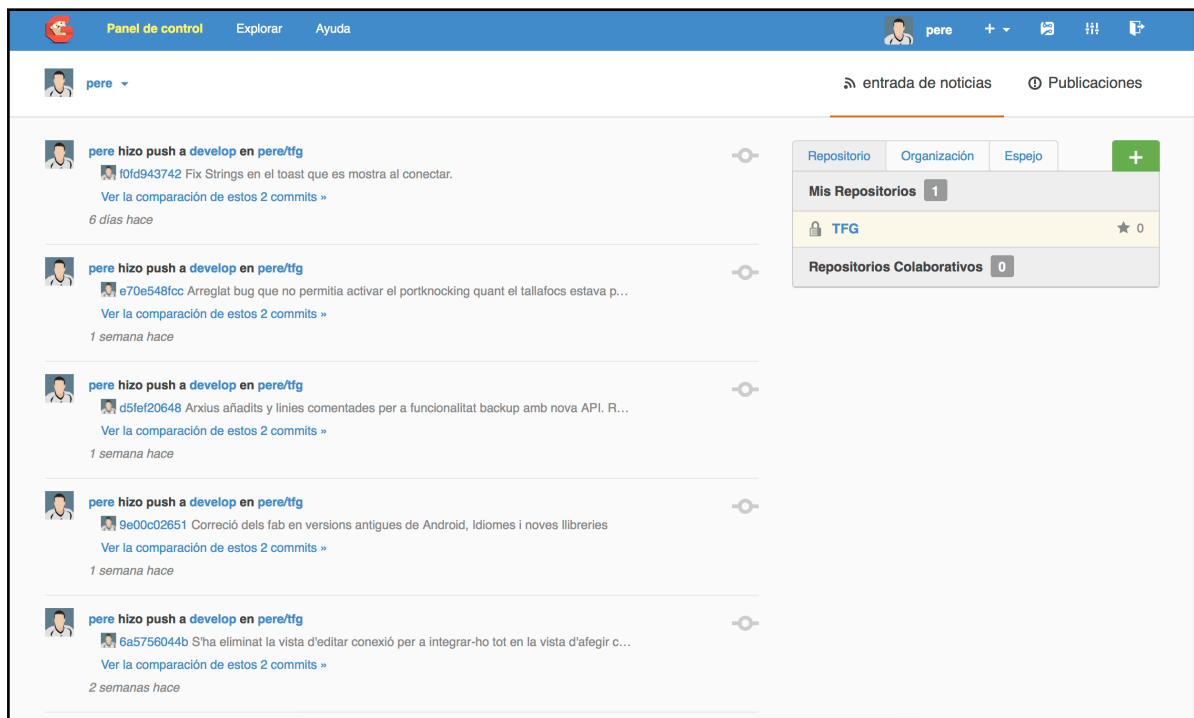
## 4.2.- Git

Git és un software de control de versions. Per a la realització del projecte es farà ús d'aquesta eina per ha mantindre un historial amb tots els canvis que vagen realitzant-se durant la realització del projecte així com la possibilitat de desfer canvis i tornar arrere a qualsevol versió anterior. El servidor el tindrà en una màquina linux externa de forma que també em servirà com a backup del projecte ja que tindrà la meua copia local i la remota. Altra opció interessant que ens proporciona Android Studio és la possibilitat de configurar aquesta eina de forma que puguem registrar tots els canvis des de el propi IDE d'una forma ràpida sense necessitat d'utilitzar els típics comandaments al terminal per a afegir els canvis. A part del Git també utilitzaré altra eina relacionada amb el Git la qual explicaré al pròxim punt com funciona.

## 4.3.- Gogs

Aquesta eina com he nomenat al punt anterior està relacionada amb Git el seu nom fa referencia a "Go Git Service" ja que és un servei per a Git i esta escrit en el llenguatge de programació Go. Gogs és un programa el qual ens permet tindre un accés web on poder controlar tots els repositoris que tenim de git així com poder vorer tots els canvis que s'han registrat entre les diferents versions d'una forma molt més amigable que simplement utilitzant git des de el propi terminal. Aquest software òbviament també el tindre instal·lat a la mateixa màquina remota que Git per a poder mostrar els repositoris.

A continuació adjunte un captura del programa.



---

## 4.4.- Java

Per a programar aplicacions per a Android encara que utilitzem Android Studio necessitarem el JDK de Java. Java és el llenguatge de programació orientat a objectes el qual utilitzarem per a programar l'aplicació.

## 4.5.- Altres eines

Les anteriors eines considere que són les més importants per a ser comentades de manera individual o no s'han comentat en algun punt anterior però òbviament s'han utilitzat moltes més com les següents:

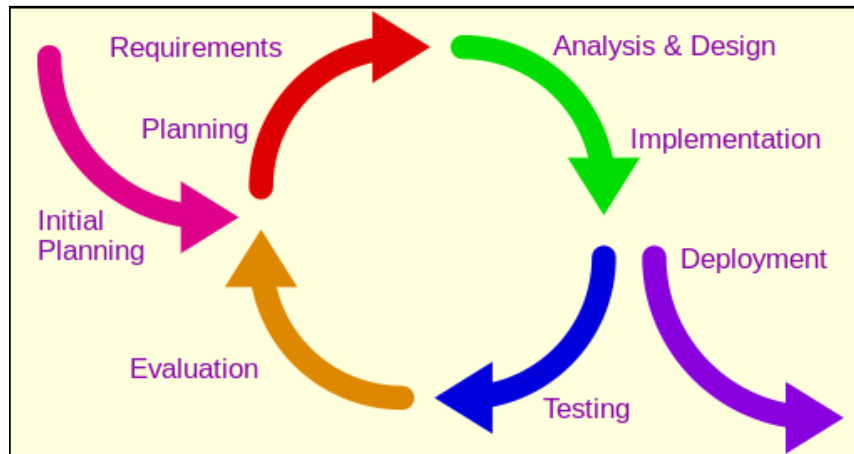
- **Adobe Photoshop:** Aquest software s'ha utilitzat per a realitzar la icona de l'aplicació.
- **Nmap:** Aquest comandament s'ha utilitzat per a comprovar si els ports estaven oberts.
- **knockd:** Servei per al servidor de port knocking.
- **iptables:** Tallafocs per a la configuració del port knocking.
- **Screeener:** Aplicació mòbil utilitzada per a la realització de captures.
- **<https://www.draw.io>:** Eina online per a la realització de diagrames UML.
- **<https://www.tomsplanner.es>:** Eina online per a la realització de diagrames de Gantt.

---

## 5.- Metodologia utilitzada

### 5.1.- Model iteratiu e incremental

Per al desenvolupament de l'aplicació mòbil s'ha elegit el model de desenvolupament iteratiu e incremental. Aquest model consisteix en realitzar el treball en xicotetes porcions que aporten una certa funcionalitat al que ja tenim creat (Incremental) de forma que sempre anem repetint el mateix procés i les mateixes pautes per a anar incloent noves millores a la nostra aplicació. Al següent diagrama es pot observar com seria el esquema de funcionament d'aquest model.



### 5.2.- Avantatges

Els avantatges del model de desenvolupament iteratiu e incremental són els següents:

- En aquest model els usuaris no han d'esperar fins que l'aplicació estiga completament finalitzada per fer ús d'ella. El primer increment compleix els requeriments més importants de tal manera que poden utilitzar l'aplicació des de la primera versió.
- En el desenvolupament d'aquest model la retroalimentació amb l'usuari es produïx molt prompte, el que permet detectar errors ràpidament i incloure millores a l'aplicació.
- Ja que els requeriments més importants es fan al principi és més difícil que a l'usuari li apleguen errors en la funcionalitat principal de la aplicació i en cas de que aplegaren es podrien detectar ràpidament.
- Permet separar la complexitat del projecte ja que primer els desenvolupa la funcionalitat principal i després s'inclouen totes les millores, intentar incloureu tot en un primera i única versió seria molt complicat i complexe.
- S'obté un aprenentatge en cada iteració que s'aplica en el desenvolupament de l'aplicació i augmenta les experiències per propers projectes, ja que al afegir cada funcionalitat o millora per separat és molt més fàcil de recordar o consultar.



---

## 5.3.- Desavantatges

Alguns dels desavantatges que té aquest model són els següents:

- Com l'entrega de l'aplicació a l'usuari es produeix molt prompte pot donar lloc a aplicacions massa simples que de vegades es veuen una mica monòtones als ulls dels usuaris que la reben.
- Requereix interacció per part de l'usuari per als següents increments, açò en alguns casos podria no ocórrer i no existir interacció per part de l'usuari..

En aquest cas aquests els desavantatges són produïts per part del usuari açò sol ocórrer en casos on l'aplicació va destinada a un client en concret. Però en el nostre cas és una aplicació per a tothom i els primers increments s'han dut ha terme mitjançant una prova pròpia de l'aplicació o deixant provar l'aplicació a diferents usuaris i preguntant-los. A més una vegada l'aplicació es publiqui al Google Play sempre tendrem un feedback amb l'usuari final mitjançant els comentaris de la pròpia tenda.

---

## 6.- Planificació

### 6.1.- Planificació per objectius

Com ja hem comentat anteriorment s'ha utilitzat un model iteratiu e incremental i per tant la millor forma de planificar el projecte és per objectius (Milestones).

El que es farà és dividir el projecte en diversos objectius que correspondrien als increments de la metodologia esmentada en el punt anterior de forma que primer creariem la funcionalitat bàsica i després afegiríem les diverses millores.

Per a realitzar l'aplicació plantejaran els següents objectius segons el seu orde:

- **Aplicació bàsica:** Aquest és el primer objectiu el qual consisteix en fer una aplicació molt simple amb una sola finestra en la qual es puguen introduir dades i enviar la seqüència de port knocking.
- **Validació de dades:** Aquí el que es farà és comprovar que les dades introduïdes per l'usuari segueixen un cert patró per a ser vàlides.
- **Funcionalitat de guardar:** A aquest objectiu afegirem la possibilitat de guardar la connexió que tenim configurada en la memòria.
- **Múltiples connexions:** A aquest punt el que es farà és afegir la possibilitat de guardar múltiples connexions mitjançant una pantalla per a introduir les dades i altra per a mostrar-les.
- **Menú per gestionar les connexions:** A aquest objectiu s'afegira un menú per a gestionar les seqüències (realitzar connexions, editar-les, elimina-les).
- **UI de l'aplicació:** En aquest punt el que es farà és adaptar el disseny de l'aplicació a l'estàndard proposat per Google que en aquest cas és Material Design.
- **Internacionalització de l'aplicació:** En aquest apartat el que es farà és traduir tots els textos de l'aplicació de forma que els tinguem en diversos idiomes.

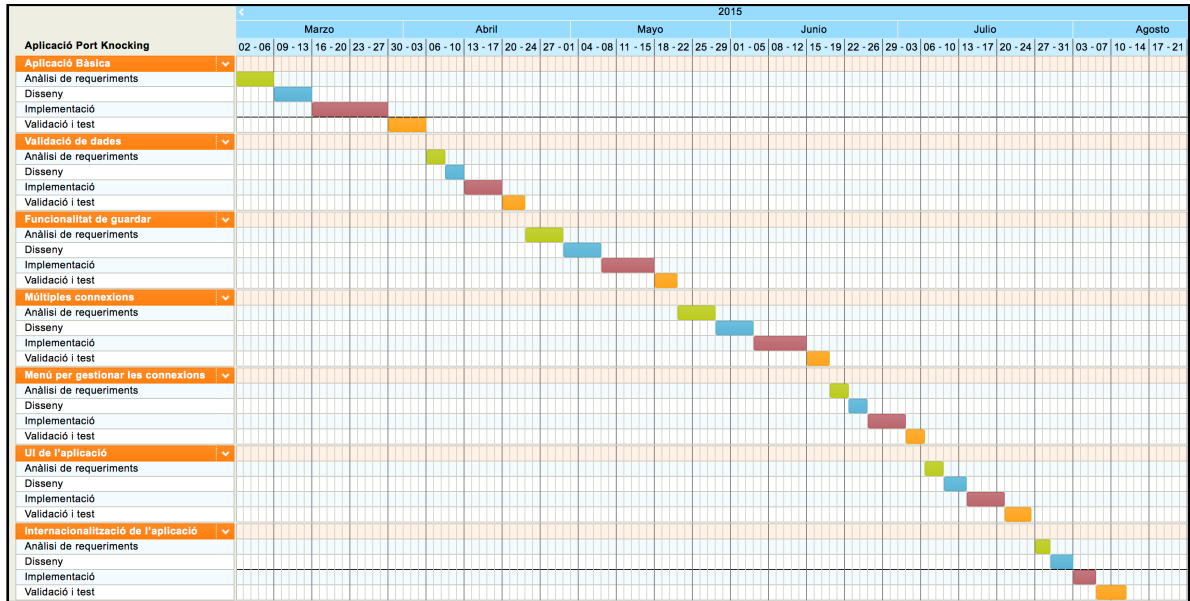
Finalment cal comentar que a cada objectiu no sols es tracta específicament de dur a terme eixe objectiu concret, sinó que a més hi ha que revisar si hi ha hagut algun error als objectius anteriors per reparar-los. També cal afegir que per a poder realitzar cada objectiu hi ha que primer analitzar-lo i dissenyar la solució i una vegada ja tinguem la solució implementar-la i testejar-la.

### 6.2.- Diagrama de Gantt

Per a planificar les dates per a la realització dels distints objectius s'ha utilitzat un diagrama de Gantt on s'ha calculat el temps estimat per a la realització de cada objectiu.

Per a planificar tots els objectius amb les diferents fases s'ha utilitzat l'eina proporcionada a la següent pagina web: <https://www.tomsplanner.es/>

Com es pot observar al diagrama a l'esquerra temin els diferents objectius amb les seues fases. Després tenim un calendari lineal on s'especifica el temps de les distintes fases de cada objectiu.



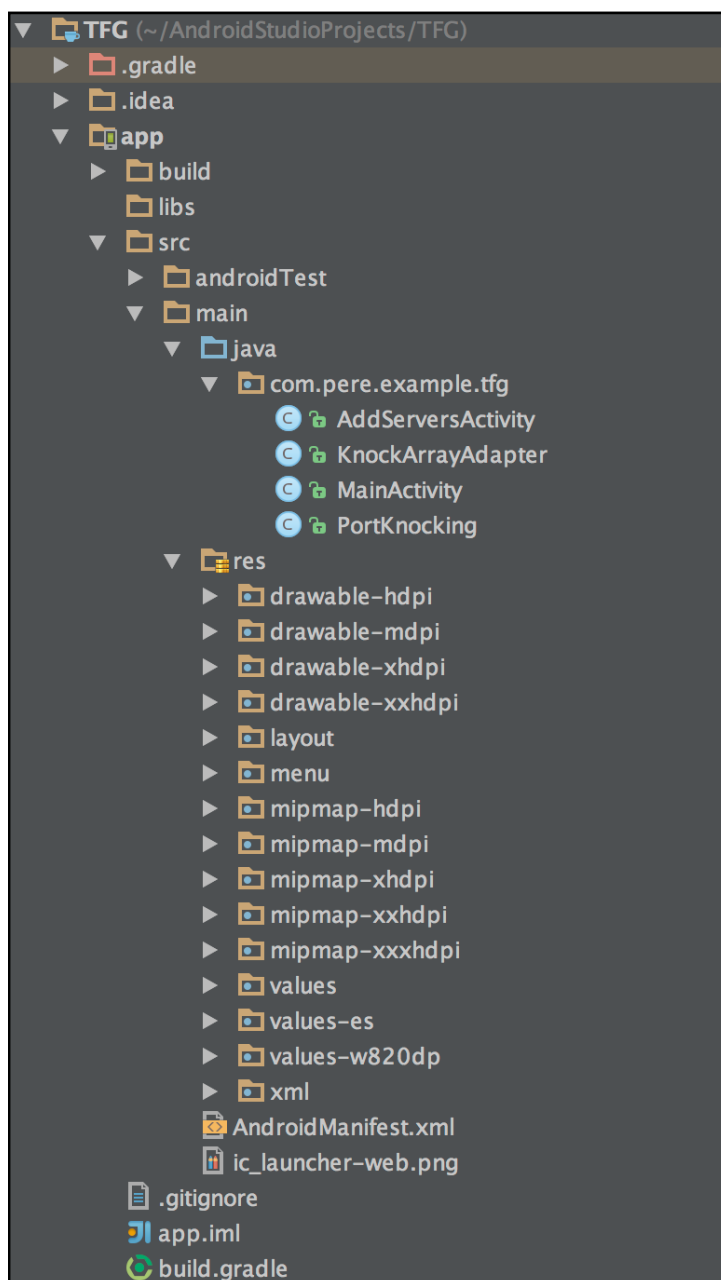
---

## 7.- Estructura i elements d'un projecte d'Android

En aquest punt vorem l'estructura i els elements més importats que formen un projecte d'Android. Aquest tipus de projectes poden ser realitzats en Android Studio o Eclipse. Segons l'IDE que utilitzem l'estructura pot variar un poc però són molt semblants. En el nostre cas com utilitzarem Android Studio per a la nostra aplicació mostrarem l'estructura d'aquest.

### 7.1.- Estructura d'un projecte d'Android

En la següent imatge del nostre projecte podem vore com s'estructura un projecte d'Android. En la imatge encara falten algunes carpetes i arxius que no apareixen encara que en aquest cas no ens interessin i sols ens fixarem en els de l'interior de la carpeta app.



---

La primera carpeta que podem veure a l'interior de la carpeta app és build. En aquesta carpeta ens interessa que al seu interior podem trobar el que són els arxius .apk (arxiu creat al compilar l'aplicació que es pot instal·lar als dispositius Android) per a trobar aquests arxius encara tindriem que accedir a la carpeta outputs i després a la carpeta apk. En la carpeta build també podem observar que es generen distints arxius R.java, aquests són els encarregats de mantindre els identificadors dels distints elements de forma que puguem accedir de forma fàcil i ràpida a ells.

La segona que trobem és la carpeta libs en la qual s'afegixen llibreries encara que actualment és molt més comú afegir-les mitjançant l'arxiu de configuració de gradle.

La següent carpeta (src) podríem dir que és la més important ja que és la que conté tot el codi. Aquí podem destacar la carpeta main que és la que conte tots els arxius i carpetes importants. En aquesta carpeta trobem la carpeta java en la qual temin tots els fitxers .java els quals contenen la funcionalitat de l'aplicació. L'altra carpeta important que trobem és la carpeta res la qual conte tots els recursos, aquests recursos els trobem organitzats en multitud de carpetes les quals contenen imatges o arxius xml. Aquests arxius xml s'utilitzen per a definir vistes, menús, colors, traduccions, animacions... Per últim a l'interior de la carpeta res cal destacar l'arxiu AndroidManifest.xml, en ell es defineix el nom de l'aplicació els tema que s'utilitzara, també definirem les Activities que tinguem i el més important de tot ací és on es defineixen els permisos que l'aplicació necessitarà.

Per últim cal destacar l'arxiu build.gradle de la carpeta src. En aquest arxiu és on es defineixen les versions mínimes de l'api, el identificador de l'aplicació. a més també tenim l'opció de dir-li quines són les llibreries que volem afegir al nostre projecte de forma que automàticament ho preparara tot per a poder utilitzar aquestes llibreries.

## 7.2.- Elements d'una aplicació d'Android

En un projecte o aplicació d'Android els principals elements que ens podem trobar són les Activities les quals representen el que serien les finestres de l'aplicació. Aquestes Activities estan formades per una part lògica seria un arxiu java i la part de la vista que seria un arxiu xml. Com podem observar a la imatge anterior també tenim un Adapter, aquest arxiu és l'encarregat de mitjançant uns valors generar els elements d'una llista i introduir els valors a una vista que podríem tindre ja predefinida la qual representaria a un únic element de la llista.

A les Activities i les vistes podem tindre multitud de elements que es poden crear i es pot accedir a ells des de qualsevol dels dos llocs. Alguns dels més importants que podem trobar segons el seu grup són els següents.

- **Layouts:** Al grup dels layouts tenim distints tipus d'elements (FrameLayout, LinearLayout, GridLayout, RelativeLayout, ...) Aquest tipus d'elements s'utilitzen per a organitzar visualment la resta d'elements quant es mostren per pantalla de forma que quede tot alineat.

- 
- **Widgets:** En el cas dels widgets tenim molts més tipus per exemple tenim els de tipus Text o TextView que són els encarregats de mostrar text per pantalla, a més també estan els del tipus Button que serien botons de distint tipus. Finalment destacarem també el Spinner, aquest element consisteix en un desplegable que pot tindre diferents valors.
  - **TextFields:** Aquest grup correspon als elements que servixen per a introduir diferents tipus de dades a l'aplicació mitjançant el teclat.
  - **Containers:** Aquest grup correspon als elements encarregats de contindre altres elements en aquest cas podem destacar elements com el ScrollView el qual és l'encarregat de que si una finestra té molta informació puguem desplaçar-nos amunt i avall. En aquest grup també tenim el ListView que s'utilitza per mostrar una llista de elements que compartixen unes certes característiques.

Per ultim cal afegir que encara existeixen alguns grups amb altres elements però com per a la realització d'aquesta aplicació no els utilitzarem no s'han nomenat.

---

## 8.- Desenvolupament de l'aplicació

En aquest punt és tractaran les diferents fases del desenvolupament de l'aplicació de forma global, és a dir, sense tindre en conte cadascun dels objectius esmentats al punt anterior

### 8.1.- Anàlisi de requisits

En aquest punt analitzarem les necessitats que té cobrir l'aplicació així com els diferents casos d'us que ens podem trobar.

#### 8.1.1.- Requisits de l'aplicació

Els requisits de la aplicació per a la aplicació que anem a desenvolupar en aquest treball són els següents:

- **Enviar peticions TCP i UDP:** Òbviament un dels requisits indispensable és poder enviar peticions ja que és la part principal de l'aplicació. A més aquest tipus de peticions es tenen que enviar indistintament per TCP o UDP ja que les seqüències poden estar formades per els dos tipus de connexions.
- **Nivell de l' API d'Android inferior:** Per tal d'aplegar al major nombre d'usuaris possible és necessari utilitzar una API que ens oferisca la major compatibilitat possible. En aquest cas després de veure el gràfic que es mostra al subapartat 3.6 d'aquest treball es poden vore els percentatges d'usuaris de cada versió d'Android amb la seua corresponent API. Per a la realització de l'aplicació s'ha optat per el nivell de l'API 14, és a dir, a partir de la versió 4.0 d'Android ja que amb aquest nivell de l'API la nostra aplicació la podria instal·lar el 95% dels usuaris d'Android.
- **Emmagatzemar les seqüències a l'aplicació:** Altre dels requisits que tenim és el d'emmagatzemar les seqüències a l'aplicació de forma que l'usuari pugua guardar-les ja que en cas de tindre diverses combinacions o combinacions molt llargues pot resultar difícil recordar-les. D'aquesta forma evitem que l'usuari pugua oblidar la seua seqüència i no pugua connectar-se al seu servidor.
- **Validació de dades:** Com l'anterior requisit aquest també serveix per a facilitar la vida a l'usuari i en cas de que intente guardar dades incorrectes es mostrara un missatge d'error avisant a l'usuari de que la informació introduïda no és correcta.
- **Gestionar les connexions o seqüències:** Altre dels requisits a complir és la gestió de les seqüències, és a dir, una vegada les tenim emmagatzemades a més de realitzar connexions ha de ser possible editar-les o eliminar-les d'una forma simple i rapida. Per a dur a terme aquesta gestió una de les millors formes és utilitzar un menú contextual el qual s'obrirà al mantindre polsada la pantalla sobre alguna de les seqüències emmagatzemades.
- **Internalització:** Actualment vivim en un món connectat on una aplicació que es publique a qualsevol tenda en un moment aplega a qualsevol punt del món

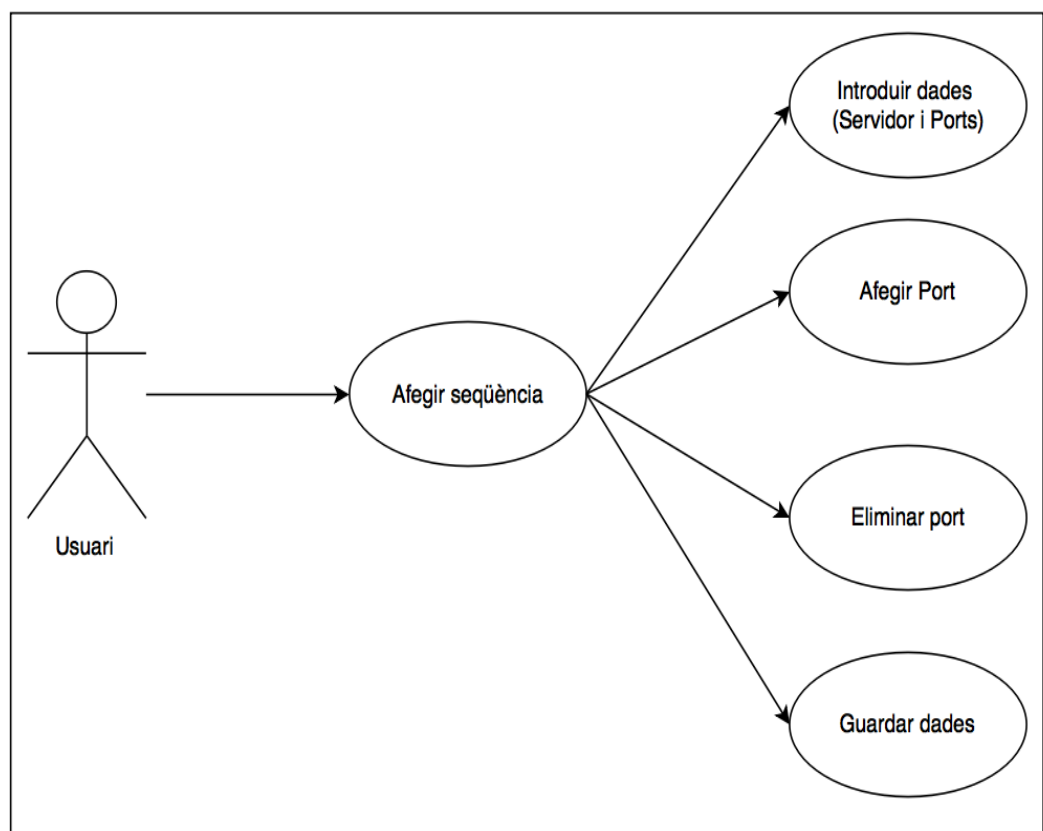
i per tant el poder tindre l'aplicació en diversos idiomes facilita l'ús a persones que viuen en altres països. Per tant el que es farà és traduir l'aplicació a l'anglès i a més l'aplicació també estarà en valencià i castellà.

- **Disseny gràfic:** Aquest punt pot paréixer menys important, però jo crec que crear un entorn que siga amigable de cara a l'usuari és un requisit important ja que l'usuari si té diverses aplicacions que realitzen la mateixa funció i complixen les seues necessitats sempre es decantara per la que siga més amigable o atractiva visualment.

### 8.1.2.- Casos d'ús

En aquest apartat es mostraran els diagrames dels diferents casos d'ús que ens podrem trobar a la nostra aplicació. Els casos que tenim són els següents:

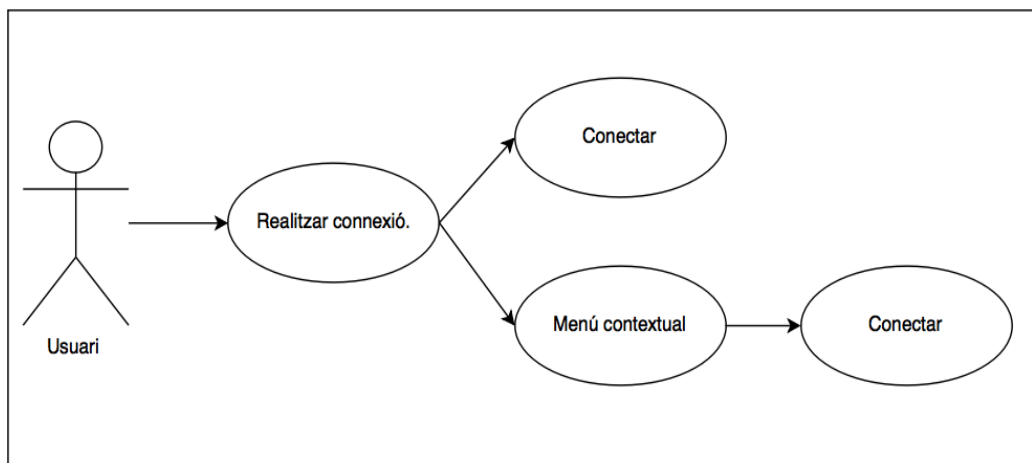
- El primer cas és el d'emmagatzemar les dades de les seqüències a l'aplicació. L'usuari tindrà una primera finestra en la qual es mostren totes les seqüències en cas d'existir-ne. L'usuari ha de seleccionar l'opció d'afegir una seqüència amb la qual li apareixerà altra finestra en la que podrà introduir les dades (servidor i ports), afegir ports per a configurar la seqüència o eliminar-los. A més tindrà l'opció de guardar la seqüència sempre i quan les dades es validen sense cap error. A continuació podem veure el diagrama d'aquest cas d'ús.





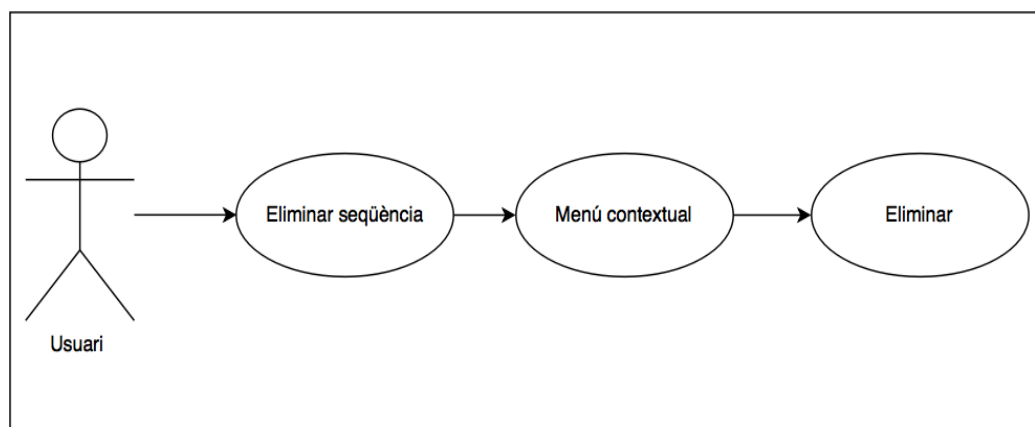
- Altre dels casos d'ús que podem tindre és en el que l'usuari ja té una una o més seqüències i vol enviar-la al servidor. En aquest cas l'usuari tindrà dues formes d'enviar la seqüència al servidor. La primera és polsant sobre l'element de la llista que pertany a la seqüència que es vol enviar. Aquesta és l'opció més ràpida, però a més tenim altra opció que és mantenir polsat sobre l'element de la llista de forma que ens apareixerà un menú contextual amb el qual podrem seleccionar l'opció d'enviar la seqüència.

A continuació podem veure el diagrama d'aquest cas d'ús.

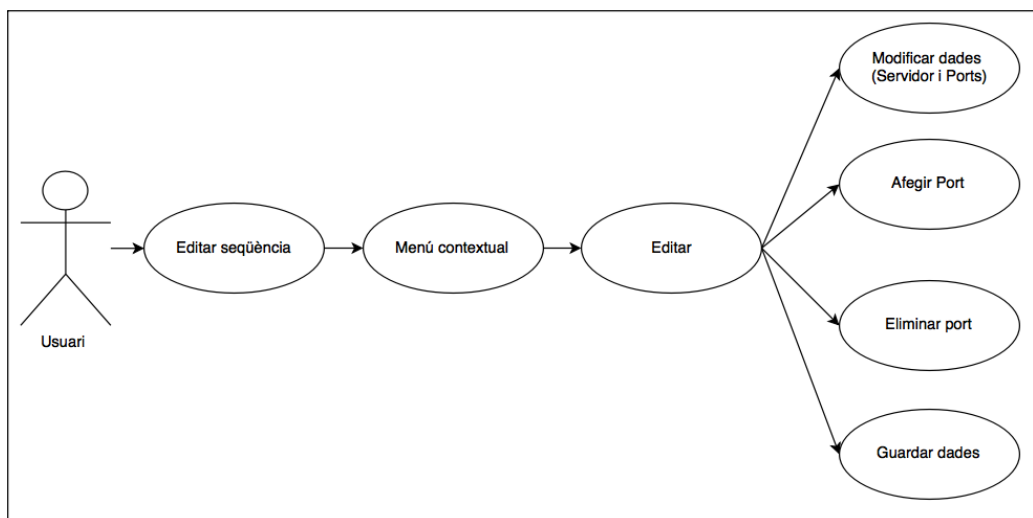


- Altre dels casos d'ús que podem tindre és en el que l'usuari ja té una una o més seqüències i vol eliminar-ne alguna. En aquest cas l'usuari tindrà una única forma d'eliminar la seqüència. Per a eliminar la seqüència l'usuari tindrà que mantenir polsat sobre l'element de la llista de seqüències de forma que li apareixerà un menú contextual amb el qual podrà seleccionar l'opció d'eliminar la seqüència.

A continuació podem veure el diagrama d'aquest cas d'ús.



- Altre dels casos d'ús que podem tindre és en el que l'usuari ja té una una o més seqüències i vol editar-ne alguna. En aquest cas l'usuari tindrà una única forma d'editar la seqüència. Per a editar la seqüència l'usuari tindrà que mantenir polsat sobre l'element de la llista de seqüències de forma que li apareixerà un menú contextual amb el qual podrà seleccionar l'opció d'editar la seqüència. Una vegada polsem aquesta opció s'obrirà una finestra com la d'afegir una seqüència però amb totes les dades de la seqüència seleccionada de forma que podrem modificar les dades o eliminar i afegir ports. A continuació podem veure el diagrama d'aquest cas d'us.



## 8.2.- Disseny

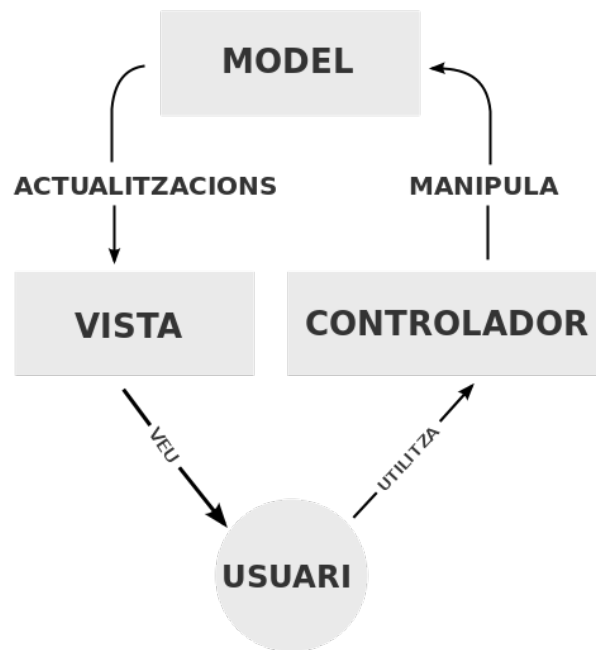
En aquest apartat de disseny de l'aplicació parlarem sobre l'arquitectura de l'aplicació, en aquest cas serà una de les més conegudes a dia de hui MVC (Model-Vista-Controlador). A més també s'explicarà quin és el diagrama de fluxe de l'aplicació.

### 8.2.1.- Arquitectura de l'aplicació

Com ja hem comentat l'arquitectura serà MVC, aquest patró de desenvolupament de programació divideix l'aplicació en tres parts interconnectades: el model de dades, l'aspecte gràfic de l'aplicació i la lògica de control.

En aquest model l'usuari utilitza el que és la part del controlador i el controlador seria el que manipula el model de dades que té. A més l'usuari veu la part proporcionada per la vista la qual és actualitzada per el model.

A continuació es pot veure un diagrama on es pot observar com està relacionada aquesta arquitectura.



En el cas d'Android tenim diversos fitxers que s'encarreguen de cadascuna de les diferents parts d'aquesta arquitectura:

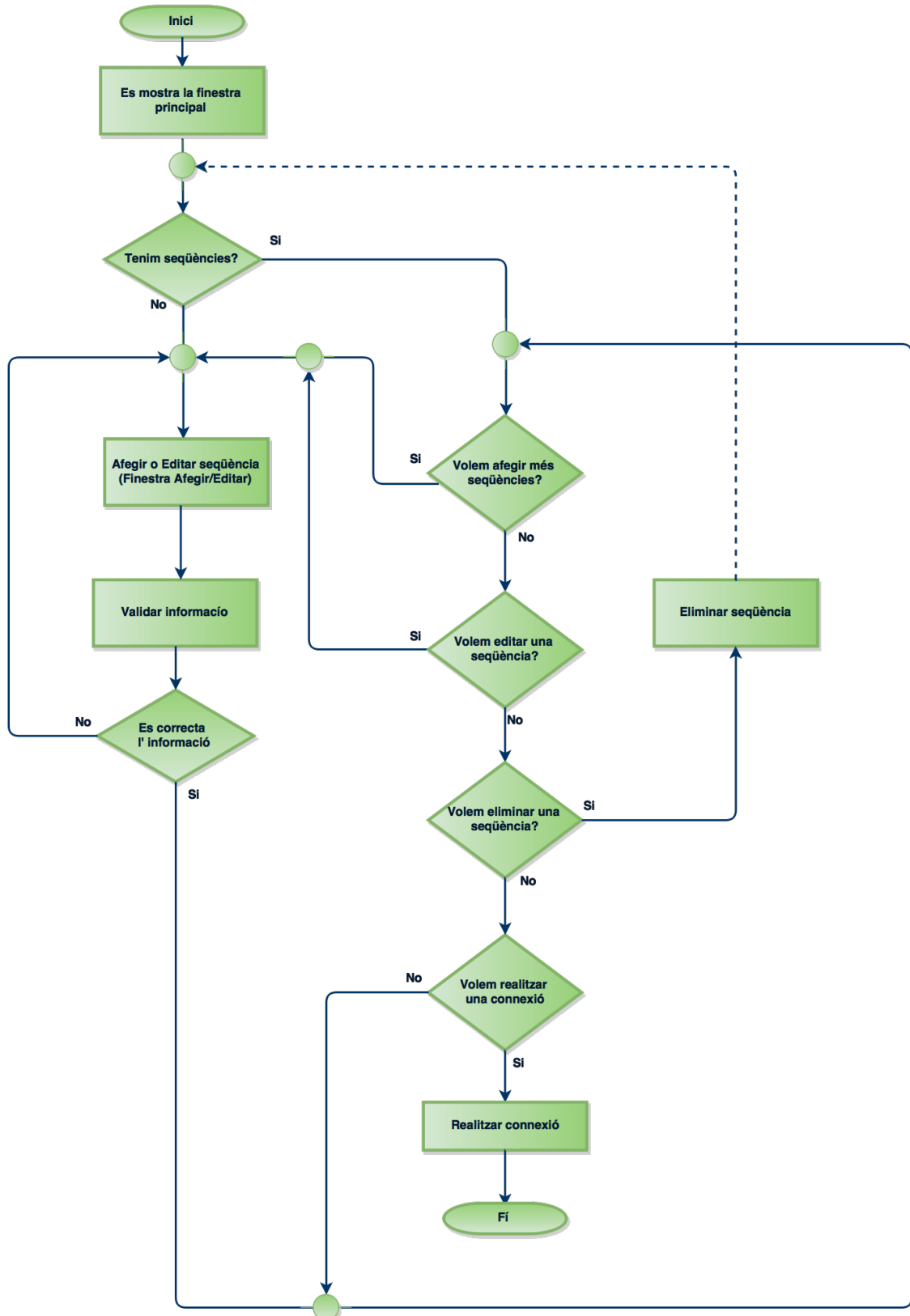
- **Model:** La part del model és l'encarregada d'emmagatzemar les dades en el cas d'Android el més comú és utilitzar SQLite, JSON i XML per a emmagatzemar les dades. En el nostre cas utilitzarem un JSON de forma que el que farem serà passar el JSON a String i emmagatzemar-lo utilitzant les preferències compartides que ens proporcionen les llibreries d'Android i ens faciliten emmagatzemar dades.
- **Vista:** La part de la vista en Android esta formada per un conjunt de arxius XML en el qual es definixen el que són els layouts mitjançant els quals es representen les distintes finestres i vistes de l'aplicació. A més aquestes vistes en XML poden ser modificades des del propi codi Java.
- **Controlador:** La part del controlador és la formada per les Activities, i fragments els quals s'encarreguen d'arreglar totes les interaccions de l'usuari de forma que per exemple detecten quan l'usuari polsa algun boto o tracten l'informació que l'usuari introduïx.

| Model  | Vista         | Controlador |
|--------|---------------|-------------|
| SQLite | XML (Layouts) | Activity    |
| JSON   | XML (Strings) | Fragment    |
| XML    | XML (Styles)  |             |
|        |               |             |

## 8.2.2.- Diagrama de fluxe

El diagrama de fluxe serveix per a representar de forma gràfica el procés del sistema, és a dir, mitjançant fletxes i símbols marcar els passos de l'aplicació.

A continuació mostrarem i explicarem el diagrama de fluxe de l'aplicació.



---

En aquest diagrama podem vore els diferents casos que podem tindre i el que podem fer segons les decisions que prenguem. En aquest cas s'ha decidit marcar el fi quan realitzem una connexió (enviar seqüència) ja que és l'objectiu al que es vol aplegar al utilitzar l'aplicació però òbviament l'usuari podrà abandonar aquesta en qualsevol moment i quan envie la seqüència també podria continuar amb l'aplicació.

Com podem observar tenim un primer cas en el que podríem tindre seqüències guardades o no, en cas de no tindre-les les crearíem sempre que superem la validació i si les tinguérem podríem realitzar més accions. Una vegada tinguem una seqüència podem afegir-ne més o editar-ne una existent de forma que s'ens enviaria a la finestra d'afegir i en el cas d'editar mostrariem les dades existents en aquesta finestra per a que es puguem modificar. Altra de les operacions que podem realitzar és la d'eliminar una seqüència, en aquest cas pot ocórrer que si sols tenim una seqüència i la eliminem ens quedem sense cap seqüència per tant en el diagrama de fluxe anirem al principi per vore si tenim connexions. Per últim si no volem realitzar cap d'aquestes accions sols ens queda l'opció d'enviar la seqüència. En cas d'enviar-la aplegaríem a la fi del diagrama i en cas de que no volguérem enviar la seqüència tornariem al principi del diagrama ja que significaria que volem realitzar altra acció.

## **8.3.- Implementació**

En aquest apartat anem a parlar de la implementació, algunes decisions que s'han pres durant aquest procés, explicarem un poc les classes i elements que tenim al projecte, explicarem un poc com s'ha realitzat i comentarem els problemes que han sorgit.

Algunes decisions com per exemple l'elecció d'Android Studio com a IDE o el nivell de l'API no els explicarem ja que s'han explicat en punts anteriors.

### **8.3.1.- Emmagatzemament de dades**

En Android per a emmagatzemar les dades tenim diferents opcions com SQLite, arxius xml, SharedPreferences...

Després d'analitzar els diferents opcions que existien es va optar per utilitzar les SharedPreferences. Aquesta classe que ens proporciona Android el que fa és emmagatzemar dades de forma que podem emmagatzemar dades utilitzant una clau i un valor.

S'ha triat aquesta opció ja que a priori en l'aplicació no s'emmagatzemaran moltes dades ja que sols hi ha que guardar les seqüències del usuari. Altra de les coses interessants d'utilitzar aquest tipus emmagatzemament és que amb la pròxima API que aplegara amb la nova versió d'Android (Marshmallow) es podrà realitzar una còpia de seguretat de les SharedPreferences i altres dades de l'aplicació que es guardaran al núvol.

---

Per a guardar tota la informació referent a les seqüències del usuari s'ha optat per utilitzar un JSON el qual guardem com a una cadena de text en les SharedPreferences. Aquest arxiu JSON té la estructura següent:

```
[{
  "name": "192.168.1.153",
  "ports": "[123, TCP], [456, UDP], [789, TCP]"
},
{
  "name": "192.168.1.153",
  "ports": "[789, UDP], [456, TCP], [123, UDP]"
}]
```

En com és pot vore en aquest exemple la estructura està formada per un JSONArray el qual conte dos elements, que serien del tipus JSON i que tindrien la informació referent a una seqüència. En el que seria la informació del que és la seqüència tindriem el nom el qual fa referència a la direcció del servidor i els ports, finalment els ports són un ArrayList que conte a la volta altres ArrayList que són els que formen el conjunt del port i el protocol.

### 8.3.2.- Implementació de la funcionalitat

Per a implementar la funcionalitat hem creat una classe anomenada PortKnocking.java, aquesta classe és l'encarregada d'enviar les peticions als diferents ports.

Aquesta classe està formada per dos mètodes un que realitza les peticions del tipus TCP i altre que realitza les del tipus UDP. Aquests dos mètodes reben com a paràmetres l'adreça del servidor i el port al qual es realitza la connexió, com a resultat d'aquestes mètodes obtenim com a resultat un Verdader o Fals per a poder saber si ha hagut algun error al realitzar la connexió. El que seria la part de desglossar cada seqüència per a cridar aquests mètodes ho faríem des de la Activity, açò s'ha fet d'aquesta manera per si algun dia volem modificar la forma de com guardem la informació ho farem a la Activity i per tant la forma de gestionar-ho canviaria però el que no variarà és com enviar una petició. Per exemple si decidirem enviar el JSON corresponent a al mètode d'enviar la connexió si algun dia per algun motiu decidirem emmagatzemar les dades a SQLite tendriem que generar un JSON o modificar les funcions.

Lo que és la part de la funcionalitat del que són els botons i la navegació de les finestres està implementada a les Activities.

En la aplicació tenim dos Activities, la que correspon a la finestra principal (MainActivity.java) i altra que correspon a la finestra d'afegir i editar seqüències (AddServersActivity.java).

En la finestra principal tenim diferents mètodes encarregats de les distintes funcions. En aquesta finestra podem destacar el mètodes per a guardar i llegir l'informació emmagatzemada a les SharedPreferences, de forma que el mètode de guardar rep un JSONArray i el guarda en les SharedPreferences i el mètode

---

de llegir quan es crida retorna un JSONArray amb la informació que hi havia guardada. També tenim un mètode encarregat de carregar tota la informació guardada i crear el ListView amb l'ajuda d'altra classe implementada al arxiu KnockArrayAdapter.java. Altre del mètodes que cal destacar d'aquesta activitat és el que implementa la funcionalitat de a partir del element seleccionat desglossar la seqüència de ports i cridar als mètodes que realitzen les connexions que hem explicat anteriorment. També podem destacar el mètode encarregat de gestionar el menú contextual del ListView el qual ens permet obrir la finestra d'editar, eliminar un element o enviar una seqüència. Finalment a la vista principal temin el mètode d'afegir una nova seqüència aquesta funció serà cridada al pulsar el botó d'afegir la seqüència de forma que ens obrirà altra finestra.

A la finestra d'editar o afegir podem trobar altres mètodes per a afegir o eliminar ports. També tenim una funció encarregada de comprovar les dades. Un dels mètodes més important que tenim a esta finestra és el de guardar el qual s'encarrega de cridar a una funció que comprova si les dades són valides després arreplega totes les dades de la vista i genera un JSON el qual es guarda mitjançant les funcions de guardar. Finalment cal destacar també que a aquesta finestra s'ha implement en la funció encarregada de carregar les dades a editar segons si al cridar a aquesta finestra se li passa algun paràmetre i en cas contrari la deixa en blanc per a afegir una nova connexió.

### **8.3.2.- Implementació gràfica de l'aplicació**

En aquest apartat parlarem de com s'ha implementat la part gràfica de l'aplicació.

Per a la implementació gràfica de l'aplicació s'han utilitzat els arxius XML on implementem les vistes. Per a aquesta aplicació encara que sols tinguem 2 Activities s'han utilitzat 4 vistes. Dos d'elles han sigut per a implementar les vistes de les Activities i les de les altres dos un representa els elements del ListView implementat en la Activity de la finestra principal i l'altre correspon a al conjunt d'elements (TextView, EditText, Spinner i botons d'afegir i eliminar) que apareixen quan afegim un port en l'altra Activity.

La finestra principal està bàsicament formada per un ListView el qual mostra totes les seqüències que tenim guardades, els elements d'aquest ListView estaran formats per altra vista ja definida en altre xml. Per últim a aquesta vista tenim el FAB (Floating Action Button) que és el botó que tenim a la part de baix a la dreta per tal d'afegir més seqüències.

La finestra d'afegir o editar ports esta formada per més elements però més simples ja que majoritàriament són elements com TextView per a mostrar textos i EditText per a rebre el contingut introduït per l'usuari. La part més destacable podria ser la que s'afegix dinàmicament que si que té alguns elements més interessants com dos FAB en la seua versió en miniatura o un Spinner, és a dir, un desplegable on poder elegir si la petició que s'enviarà al port serà TCP o UDP.

Durant implementació de l'apartat gràfic s'ha tingut en conter les normes de disseny proposades per Google amb Material Design. Material Design és una

---

normativa de disseny enfocada en la visualització del sistema operatiu Android, a més de en la web i en qualsevol altra plataforma. Comparat amb el disseny anterior que existia a Android Material Design proposa un disseny més net, en el qual predominen animacions i transicions, els colors i els efectes de profunditat tals com la il·luminació i ombres. Google ens proporciona una web on explica com aplicar Material Design.

<https://www.google.com/design/spec/material-design/introduction.html>

En aquesta web Google ens recomana utilitzar les paletes de color que ens proporciona en la seua web. En les següents pàgines web podem trobar la paleta de colors recomanada en la web de Google i en GitHub un XML el qual em utilitzat durant implementació per tal de tindre definides aquestes paletes de color.

<https://www.google.com/design/spec/style/color.html#>

<https://gist.github.com/daniellevass/b0b8cfa773488e138037>

Un dels elements que podem destacar d'aquest tipus de disseny és el FAB el qual és un tipus de botó flotant de forma circular el qual és molt comú en aquest tipus de disseny. Per a implementar aquest tipus de botó s'ha utilitzat una llibreria externa ja que la pròpia de Google donava problemes de compatibilitat en versions anteriors d'Android, la llibreria que s'ha utilitzat es pot trobar a GitHub on explica com configurar-la amb l'arxiu build.gradle. El enllaç és el següent:

<https://github.com/Clans/FloatingActionButton/>

Per últim a l'aplicació també s'ha tingut que implementar una icona la qual no s'ha realitzat en Android Studio ja que no té aquest tipus de opcions. Per al que són imatges s'ha utilitzat Photoshop, per a la realització de la icona també s'han seguit les normes de disseny proposades utilitzant colors de la paleta i elements que tenen una elevació amb una il·luminació que crea ombres en la icona. Com es pot veure el resultat és el següent:



La icona de l'aplicació en aquest cas representa una mà colpejant sobre el fons de la icona. El motiu de aquesta icona és per la referencia a port knocking (colpejament de ports)



---

## 8.4.- Validació i test

Per a comprovar que tot és correcte després de cada increment o millora en la aplicació s'ha comprovat la millora introduïda però en tots els casos s'ha comprovat que la funcionalitat bàsica seguia funcionant, és a dir, s'han comprovat que les seqüències s'envien sense cap problema per això s'ha utilitzat el knockd en mode verbose en el qual podem observar les peticions que apleguen al servidor.

```
pere@linux-mint ~ $ sudo knockd -v
[sudo] password for pere:
listening on eth0...
192.168.1.102: openSSH: Stage 1
192.168.1.102: openSSH: Stage 2
192.168.1.102: openSSH: Stage 3
192.168.1.102: openSSH: OPEN SESAME
openSSH: running command: /sbin/iptables -I INPUT -p tcp --dport 22 -j ACCEPT -m comment --comment "Regla Port Knocking SSH"
```

A més d'aquest tipus de test una vegada ja teníem l'increment amb el qual incloem les ListView amb elements guardats també s'ha comprovat que si tenim seqüències guardades estes es mantinguen intactes després de realitzar diferents operacions sobre altres com per exemple eliminar una de totes les seqüències que tinguem.

Aquest tipus de proves s'ha realitzat a distintes versions d'Android per comprovar que funcionava bé a les distintes versions entre 4.0 i 6.0. Per a fer aquestes comprovacions s'ha fet us de l'emulador proporcionat per Android Studio en el qual podem configurar distints dispositius amb diferents versions d'Android i diferents característiques.

Finalment comentar que també s'han realitzat test d'usabilitat, per a valorar aspectes com el temps requerit per l'usuari per a dur a terme una tasca o els errors que realitza abans d'aplegar a l'objectiu. Després d'aquest test és va tindre que revisar la pantalla d'afegir connexions ja que en un principi tenia un FAB com el de la finestra principal per a afegir ports, però açò era un problema ja que cada vegada que introduíem dades als camps el boto pujava o baixava al aparèixer el teclat i si es posava fixe quedava ocultat per el teclat. A més aquest botó afegia un port més al final de la llista i per tant si ens equivocàvem i volíem afegir un port en mig d'altres teníem que eliminar els valors dels camps per a posar-lo en el seu lloc. Per a arreglar açò es va dissenyar una nova finestra la qual té dos FAB en la seua versió mini de forma que estan fixes junt amb el port de forma que ens permeten eliminar el port o afegir un port en la posició següent.

## 8.5.- Problemes trobats

Durant el desenvolupament de l'aplicació han sorgit alguns problemes com el problema de disseny comentat anteriorment i detectat gracies al test d'usabilitat. Però a més han sorgit altres problemes. Per exemple la llibreria encarregada d'implementar el FAB proporcionada per Google donava problemes en versions

---

prèvies d'Android i amb dispositius amb la pantalla menuda ja que els FAB en la seua versió mini tenien una separació anormal i per tant altres elements de la UI apareixien fora de la seua posició. En quant a les versión prèvies d'Android també va sorgir el problema de que en els EditText el color de la lletra apareixia en blanc en conte de negre com en versions més actuals, açò es va poder arreglar forçant per codi a que el EditText tinguera assignat el color negre.

En quant als nivells de l'API van aparéixer problemes de compatibilitat però utilitzant els mètodes específics de l'API antiga es solucionaven. El que si que em va cridar l'atenció és que el nivell de l'API utilitzada no fora compatible amb el mètode remove dels JSONArray, la forma de solucionar aquest problema va ser eliminar l'element de forma manual iterant tots els elements del JSONArray i creant un nou JSONArray sense el element a eliminar.

També podem destacar un xicotet problema a les traduccions ja que s'ha configurar per defecte l'anglés i s'ha afegit les traduccions al castella i al català o valencià. El problema que ocorre és que si un usuari configura el mòbil en gallec o basc l'aplicació apareixerà en anglés. Per tal d'evitar aquesta situació simplement s'ha afegit les traduccions del castellà a aquestes idiomes fins que tinga la possibilitat de contactar en alguna persona coneixedora d'aquestes llengües per tal de traduir l'aplicació. En cas d'algun usuari que utilitze alguna llengua d'altre país (italià, francès, alemà...) se li mostrara l'aplicació en anglés.

Per últim comentar que la part referent a les seqüències també ha donat problemes ja que segons la configuració del tallafocs (DROP, REJECT...) les peticions enviades per TCP o UDP donaven excepcions distintes segons cada cas i per tant s'han tingut que controlar per a vorer si realment la seqüència s'havia enviat correctament o si per el contrari s'havia produït algun error.

---

## 9.- Manuals de l'aplicació

### 9.1.- Manual d'instal·lació



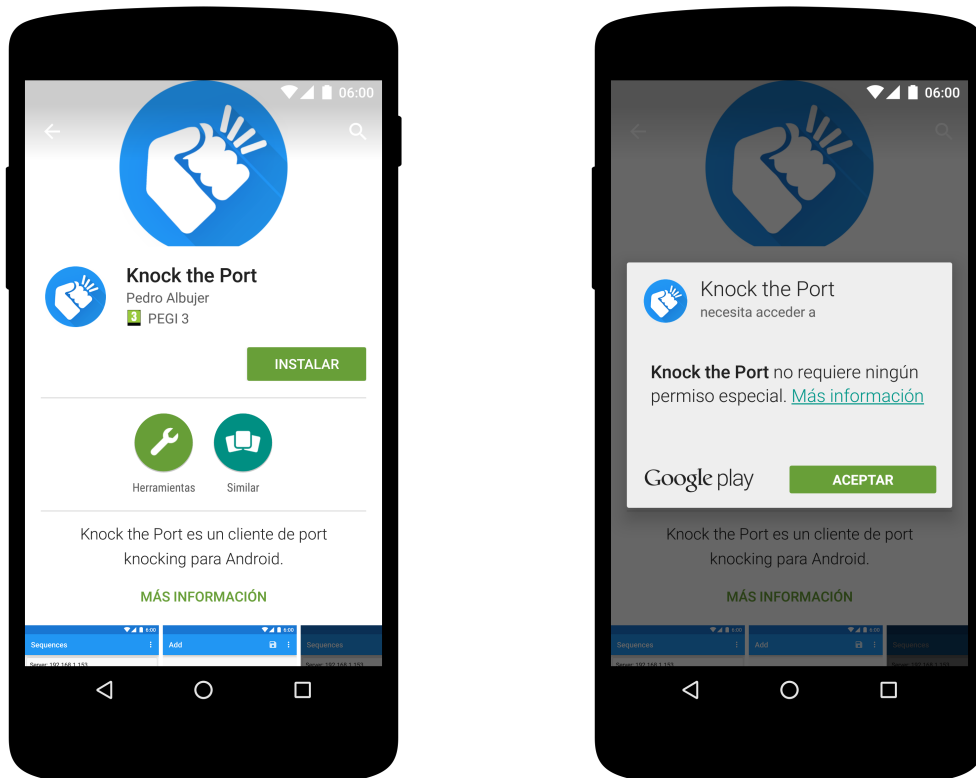
La majoria de dispositius Android porten instal·lada de fàbrica una versió de l'aplicació Google Play, la botiga oficial d'aplicacions Android de Google. Aquesta aplicació permet descarregar i instal·lar un moltíssimes d'aplicacions per a aquest sistema operatiu d'una manera senzilla i còmoda.

A més l'usuari pot instal·lar directament l'aplicació que desitge al seu dispositiu Android de dues maneres, buscant-la en l'aplicació Google Play o bé a la web oficial d'aquest servei.

El procés és tan simple com buscar "Knock the Port" o "Port knocking" a Google Play i seleccionar l'aplicació i després en la finestra on tenim tots els detalls sobre l'aplicació pulsar sobre el boto instal·lar. També podem accedir a l'aplicació mitjançant el següent enllaç:

<https://play.google.com/store/apps/details?id=com.albujer.pere.knocktheport>

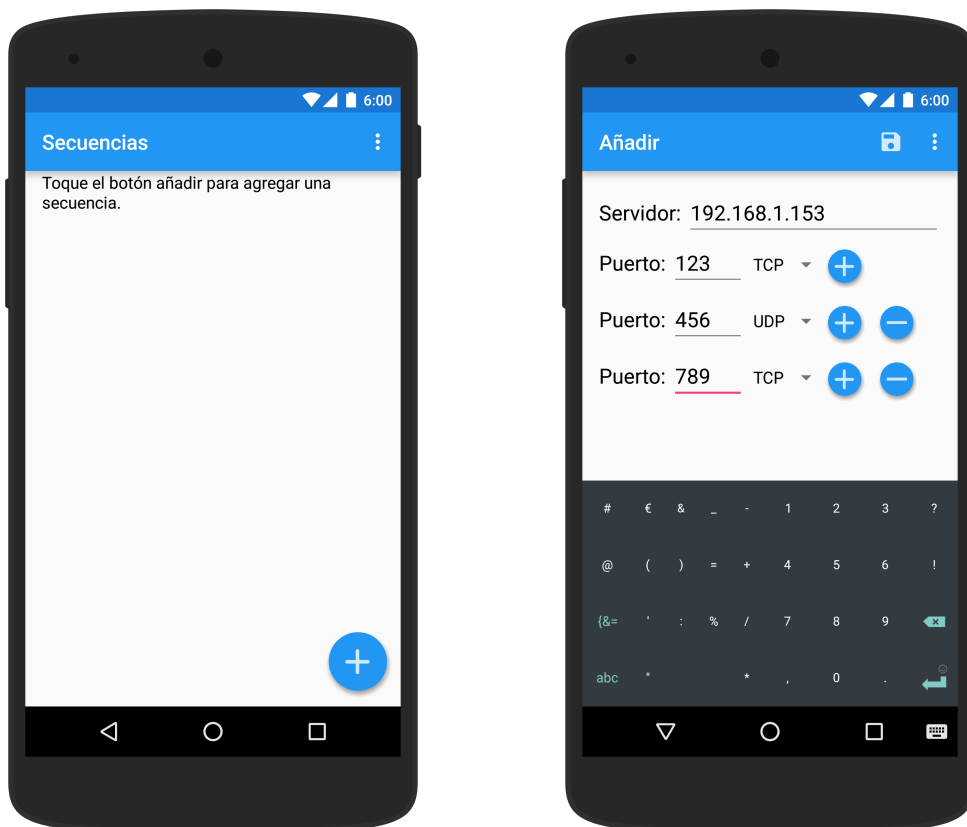
A continuació podem veure unes imatges del procés d'instal·lació:



## 9.2.- Manual d'ús

L'ús d'aquesta aplicació resulta molt senzill per a qualsevol usuari ja que és similar a la majoria d'aplicacions d'Android, però, és convenient dictar algunes pautes i passos per a que tots els usuaris puguin adaptar-se amb facilitat al conjunt d'opcions que ofereix.

En primer lloc una vegada instal·lem l'aplicació ens trobem amb aquesta finestra (imatge de l'esquerra). Per a poder afegir una seqüència tindrem que polsar el botó blau de la part inferior dreta el qual ens portarà a una nova finestra (imatge de la dreta).

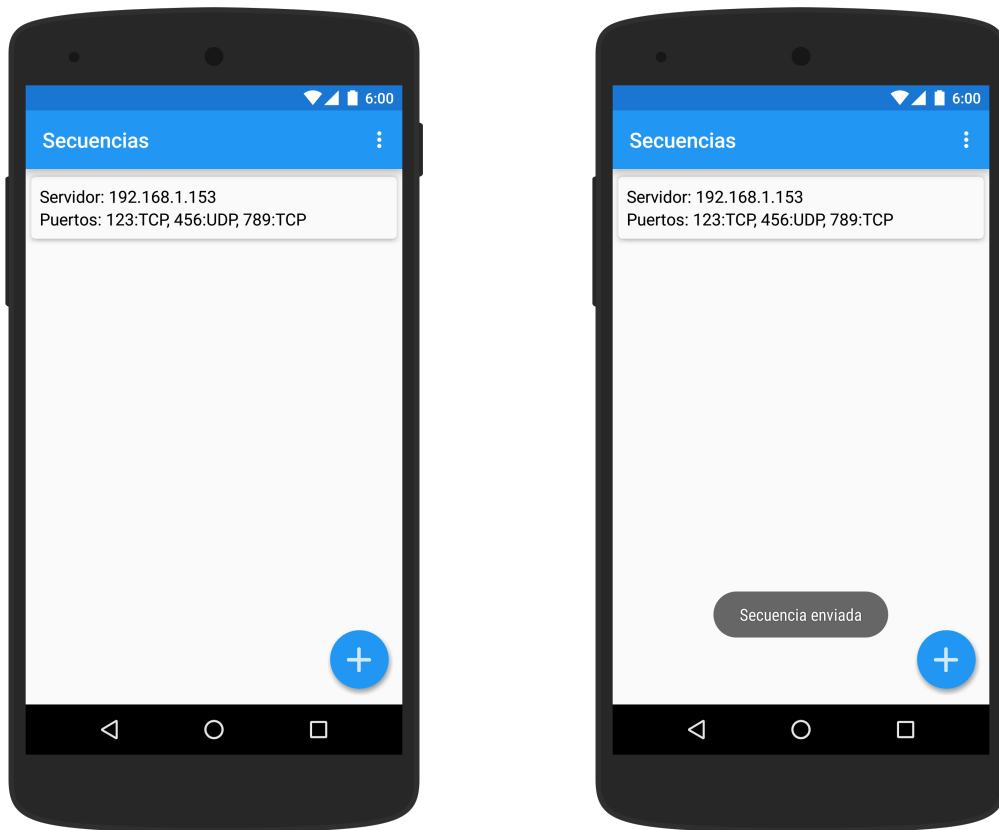


Com es pot observar en la imatge de la dreta en aquesta finestra podem afegir la nostra seqüència. Amb el botó + afegirem un nou port a continuació i amb el botó - eliminarem el port. Una vegada la seqüència estiga configurada la podem guardar mitjançant el botó que tenim en la barra superior amb l'icona de guardar, en cas de no voler la guardar-la podem tornar arrere amb el botó arrere del nostre mòbil.

Una vegada em guardat la seqüència automàticament tornarem a la primera finestra on ja podem vore la seqüència que acabem de guardar a no ser que existisca alguna dada invalida i per tant es mostrara un error en la finestra d'afegir. Una vegada ja tinguem la seqüència a la llista sols hem de polsar sobre ella per tal d'enviar les peticions al servidor. En la següent imatge (imatge de l'esquerra)

---

podem veure la seqüència a la llista i a l'imatge de la dreta podem veure el missatge que es mostra en cas d'enviar la seqüència de manera satisfactòria.



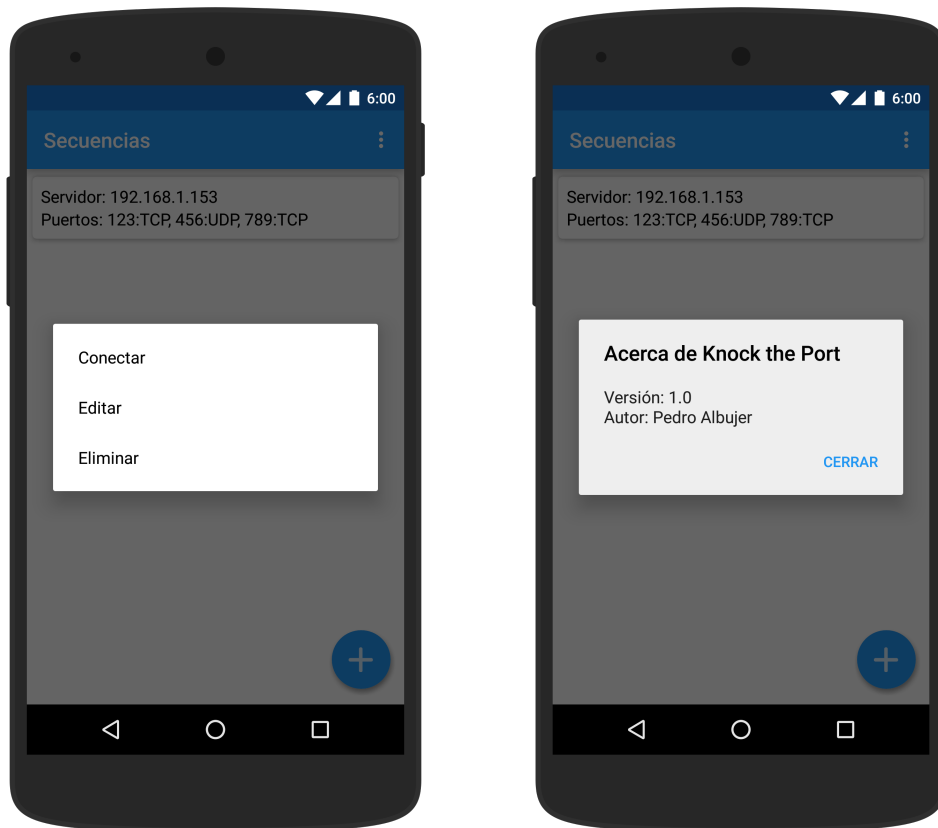
Si en conte pulsar de manera normal sobre la seqüència de la llista el que fem és fer-ho de manera prolongada ens apareixerà un menú amb les següents opcions connectar, editar i eliminar. La primera de les opcions en servirà per a poder enviar la seqüència al igual que si polsem sobre ella a la llista. L'opció d'editar ens permetrà modificar les dades de la seqüència seleccionada de forma que ens apareixerà una finestra com la d'afegir en la qual podrem modificar les dades. Per últim també podrem eliminar una seqüència des de aquest menú, en cas de seleccionar aquesta opció la seqüència s'eliminarà i s'actualitzarà la llista de seqüències.

Si volguérem eixir del menú simplement polsant el botó arrere tornariem a la pantalla de seqüències. A més també tenim altre popup en la qual es mostra informació de l'aplicació en el qual ocorre el mateix per a eixir, però a més tenim el botó de tancar.

Per últim mostrarem una imatge (imatge de l'esquerra) corresponent al menú que apareix al mantenir polsat sobre un seqüència i també mostrarem el popup amb la informació de l'aplicació el qual apareix al pulsar sobre la icona de menú de la part

---

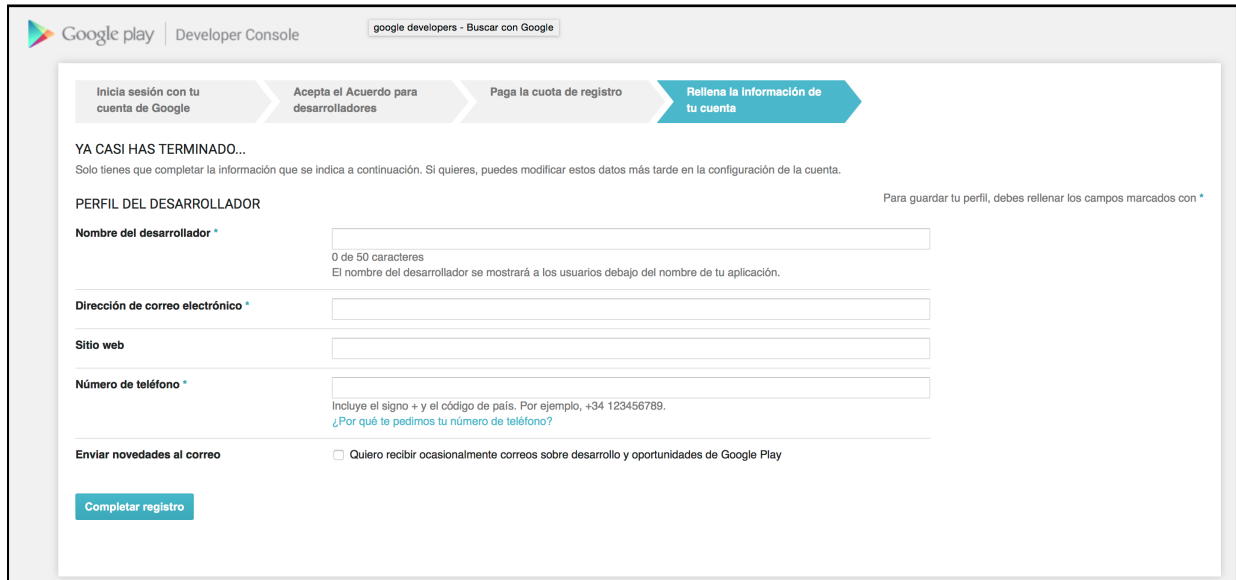
superior dreta i seleccionar l'opció per a poder vore les dades de l'aplicació (imatge de la dreta).



## 10.- Publicació a Google Play

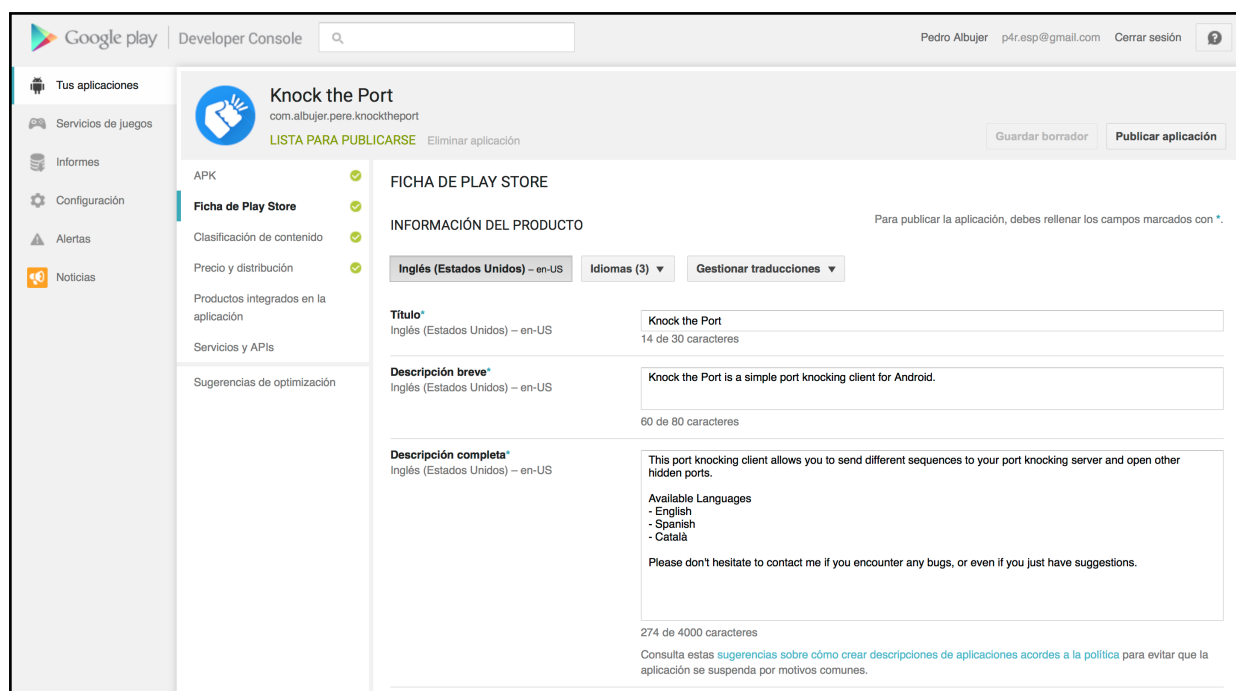
Google Play és la major tenda d'aplicacions d'Android així que per tant distribuïrem de manera gratuïta l'aplicació mitjançant aquesta plataforma. Els passos que cal seguir per a pujar l'aplicació són els següents:

- Registrar-se a la següent pàgina web <https://play.google.com/apps/publish/signup/> i pagar un xicoteta quota de 25\$.



The screenshot shows the 'Rellena la información de tu cuenta' step in the Google Play Developer Console registration process. It includes a progress bar at the top with four steps: 'Inicia sesión con tu cuenta de Google', 'Acepta el Acuerdo para desarrolladores', 'Paga la cuota de registro', and 'Rellena la información de tu cuenta'. Below the progress bar, there is a message: 'YA CASI HAS TERMINADO... Solo tienes que completar la información que se indica a continuación. Si quieres, puedes modificar estos datos más tarde en la configuración de la cuenta.' The main section is titled 'PERFIL DEL DESARROLLADOR' and contains several form fields: 'Nombre del desarrollador \*' (0 de 50 caracteres), 'Dirección de correo electrónico \*', 'Sitio web', and 'Número de teléfono \*'. There is also a checkbox for 'Enviar novedades al correo' and a 'Completar registro' button at the bottom.

- Una vegada ja registrat hem de seleccionar l'opció de publicar una aplicació i ens apareixerà una finestra en la qual tendrem que completar amb l'informació de l'aplicació. Com a curiositat podem destacar que fins que no va aplegar aquest moment no es va tindre en conte el nom de l'aplicació i per tant en eixe mateix moment vaig buscar algun nom que no existira i el qual representara a l'aplicació. El nom elegit és "Knock the Port" el que podríem traduir com "Colpeja el Port"



The screenshot shows the 'Knock the Port' application page in the Google Play Developer Console. The application is listed as 'LISTA PARA PUBLICARSE'. The page includes a sidebar with navigation options like 'Tus aplicaciones', 'Servicios de juegos', 'Informes', 'Configuración', 'Alertas', and 'Noticias'. The main content area is titled 'FICHA DE PLAY STORE' and contains 'INFORMACIÓN DEL PRODUCTO'. It includes fields for 'Título' (Knock the Port), 'Descripción breve' (Knock the Port is a simple port knocking client for Android.), and 'Descripción completa' (This port knocking client allows you to send different sequences to your port knocking server and open other hidden ports.). There are also buttons for 'Guardar borrador' and 'Publicar aplicación'.

---

## 11.- Open Source

Una vegada ja realitzada l'aplicació m'he donat conte de la quantitat de llibreries que existeixen per a Android compartides de forma lliure, les quals permeten crear aplicacions d'una manera més ràpida i simple. Aquest és uns del motius per els quals he decidit fer public el codi de la meua aplicació de forma que si algú vol utilitzar-lo per a algun altre projecte o per a qualsevol cosa pugua fer-ho.

### 11.1.- Llicencia

En quant a la llicencia és una de les coses que m'ha costat elegir i per tal de prendre aquesta decisió me ajudat de la següent pagina web proporcionada per GitHub: <http://choosealicense.com>

Després de vorer els distints tipus de llicencies amb les seues característiques m'he decidit per elegir la MIT License.

La llicència MIT és una llicència permissiva la qual és curta i directa. La qual permet a la gent fer el que vulguen amb el codi, sempre que proporcionen l'atribució de nou a l'autor i no el facen responsable.

La llicencia és la següent:

The MIT License (MIT)

Copyright (c) 2015 Pedro Albuja

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

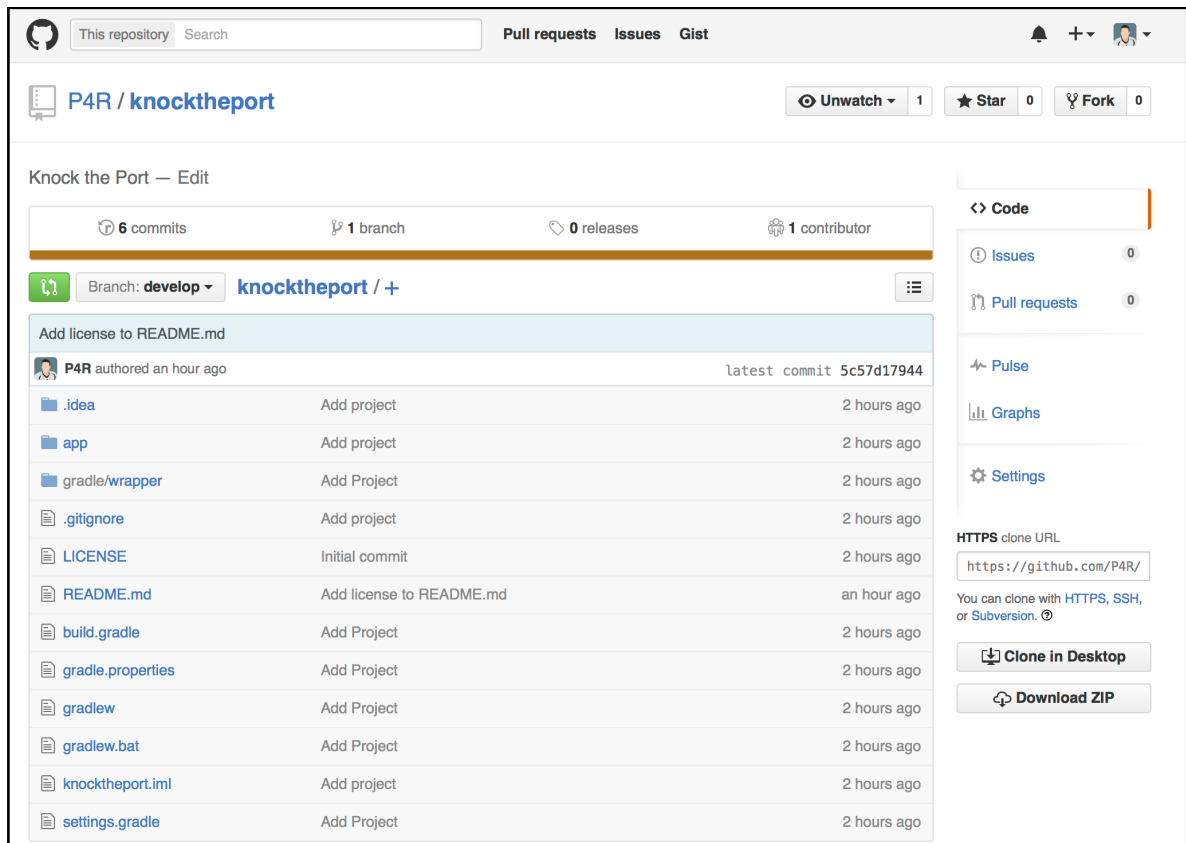
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



## 11.2.- Publicació a GitHub

GitHub és un servei de hosting de repositoris Git, el qual ofereix tota la funcionalitat de Git de control de revisió distribuït i administració de codi de la font, afegint les seves característiques pròpies. A diferència de Git, GitHub proporciona una interfície gràfica basada en web. També proporciona control d'accés i diverses característiques de col·laboració com bug tracking, administració de tasques, i wikis per cada projecte. En la següent imatge es pot veure el repositori de GitHub de l'aplicació.



The screenshot shows the GitHub interface for the repository 'P4R / knocktheport'. At the top, there's a search bar and navigation links for 'Pull requests', 'Issues', and 'Gist'. Below that, the repository name is displayed along with 'Unwatch', 'Star', and 'Fork' buttons. The main content area shows the repository's metadata: 6 commits, 1 branch, 0 releases, and 1 contributor. A commit history table is visible, listing files added or modified, such as '.idea', 'app', 'gradle/wrapper', '.gitignore', 'LICENSE', 'README.md', 'build.gradle', 'gradle.properties', 'gradlew', 'gradlew.bat', 'knocktheport.iml', and 'settings.gradle'. On the right side, there are links for 'Code', 'Issues', 'Pull requests', 'Pulse', and 'Graphs', along with a 'Settings' button and a 'Clone in Desktop' button.

Per a poder publicar el projecte a GitHub necessitem tindre un compte a la web. En el meu cas ja en tinc un però es pot crear de manera gratuïta des de el següent enllaç: <https://github.com/join>

Una vegada tenim el compte i hem entrat amb ell en la part superior dreta tenim un botó amb el símbol + amb el qual podrem crear un nou repositori. Una vegada li donem a crear podrem indicar el nom del repositori i a més en donara l'opció d'afegir diversos fitxers. Un dels fitxers és el README.md en el qual podem explicar un poc el projecte, altre fitxer és el .gitignore en el qual podem seleccionar la configuració del tipus de projecte de forma que s'ignoren algun tipus de fitxer a l'hora de pujar el projecte com per exemple els arxius compilats, finalment també podrem seleccionar el tipus de llicència de forma que ens afegira el arxiu necessari.

---

Per a poder poder pujar el projecte simplement tindrem que clonar el repositori al nostre ordinador mitjançant el comandament “git clone” en el qual li passarem com argument la url del repositori. Una vegada ja tinguem el repositori simplement copiarem o mourem el projecte a eixa carpeta i tornarem a pujar els canvis al servidor de GitHub. La seqüència de comandaments seria la següent:

```
git clone https://github.com/P4R/knocktheport.git
mv TFG/* knocktheport
cd knocktheport/
git add .
git commit -m "Add project"
git push origin master
```

Una vegada realitzats aquest comandaments ja tendriem el nostre repositori en GitHub. Quan pugem els canvis al servidor per primera vegada al executar el comandament “git push origin master” ens demanara els nostre usuari i contrasenya de GitHub per a que sols nosaltres puguem pujar canvis a no ser que li donem permisos a alguna altra persona.

Per a accedir al repositori i al codi del projecte ho podem fer des del següent enllaç:

<https://github.com/P4R/knocktheport>

---

## 12.- Possibles ampliacions

### 12.1.- Ampliacions en procés

En aquest apartat comentarem una de les ampliacions que està en procés però la qual encara no s'ha portat a terme de manera definitiva ja que és una millora proporcionada per la nova API la qual ens dona la possibilitat de amb la nova versió d'Android realitzar còpies de seguretat de les SharedPreferences les quals es guardaran al Google Drive de l'usuari sense ocupar l'espai que l'usuari té per a documents. Per a dur a terme aquesta implementació s'ha estat provant amb la versió de prova de l'API i amb la última versió de prova d'Android. Una volta creats els arxius necessàries i s'ha afegit alguna línia al AndroidManifest.xml i s'ha provat aquesta nova funcionalitat finalment s'ha comentat aquesta part del codi per tal de poder seguir utilitzant les versions antigues de l'API amb la seua compatibilitat amb versions anteriors.

### 12.2.- Ampliacions futures

Com a ampliacions futures m'agradaria adaptar l'aplicació de forma que fora compatible amb Android Wear, aquest sistema operatiu és el que Google utilitza als rellotges intel·ligents. La idea d'adaptar l'aplicació a aquest sistema és que l'usuari una vegada tinga configurades les seqüències al seu dispositiu mòbil pugui enviar les peticions mitjançant el rellotge de forma que no necessite traure el mòbil de la butxaca.

---

## 13.- Conclusions

Per acabar i una vegada acabada tant la redacció d'aquest document escrit com el desenvolupament de l'aplicació objectiu d'aquest treball, es poden extreure diverses conclusions interessants que posen de manifest l'estudi dut a terme per assolir els objectius proposats a l'inici i que sens dubte ajudaran a construir una perspectiva més precisa a l'hora de començar un altre projecte de complexitat similar.

El primer propòsit que s'ha perseguit ha sigut conèixer en profunditat el sistema operatiu sobre el qual s'executa l'aplicació desenvolupada. Per això s'ha realitzat un anàlisi de la seua arquitectura interna, dels elements que constitueixen un projecte i dels components que constitueixen les aplicacions.

Un altre objectiu aconseguit durant aquest desenvolupament ha sigut la construcció d'una UI seguint la filosofia de disseny centrat en l'usuari i les guies de Material Design, on l'important és crear un producte visual e intuïtiu que aporte una experiència satisfactòria als usuaris, aconseguint que aquests realitzen les tasques amb el mínim esforç possible.

També s'han de destacar les tècniques d'enginyeria del software utilitzades en l'anàlisi i disseny previs a la implementació, les quals, indubtablement, ajuden molt a fixar els objectius i requisits que ha de complir el projecte, el que permet abordar les fases de desenvolupament amb unes metes específiques i plantejades prèviament.

Des del meu punt de vista personal he de dir que m'ha sigut molt agradable el poder desenvolupar un projecte software d'aquestes característiques, a més he de dir que és molt gratificant una vegada acabada l'aplicació el poder compartir-la amb tot el món mitjançant la publicació al Google Play. Per últim afegir que aquest projecte m'ha permès posar en pràctica molts dels conceptes adquirits al llarg dels diferents cursos d'aquest grau. És en aquests moments quan un realment es dona compte de la importància de cadascuna de les experiències viscudes a la universitat i de les dificultats superades.

---

## 14.- Bibliografia

<http://developer.android.com/reference/packages.html>

<http://stackoverflow.com>

<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

[https://es.wikipedia.org/wiki/Golpeo\\_de\\_puertos](https://es.wikipedia.org/wiki/Golpeo_de_puertos)

<https://es.wikipedia.org/wiki/Android>

<https://developer.android.com/about/dashboards/index.html>

[https://es.wikipedia.org/wiki/Desarrollo\\_iterativo\\_y\\_creciente](https://es.wikipedia.org/wiki/Desarrollo_iterativo_y_creciente)

<https://ca.wikipedia.org/wiki/Model-View-Controller>

<https://ca.wikipedia.org/wiki/GitHub>