

Manual de compilació de l'aplicació

Martí Masot

Índex

Índex	1
Requisits	2
Instal·lació i preparació del software necessari	3
NVM i Node.js.....	3
Android Studio Command Tools.....	4
Java JDK 17.....	6
Descàrrega del codi font i dependències.....	7
Claus i autoritzacions necessàries.....	8
Compilació de l'aplicació	9
Web.....	10
Tunneling amb Cloudflare.....	11
Android.....	12

Requisits

Especificació	Requisits mínims	Requisits recomanats
CPU	2 nuclis a 1.2 GHz	4 a nuclis a 1.8 GHz
RAM	2 GB	4 GB
HDD	10 GB	25GB
SO	Debian 10-11	Debian 12

Instal·lació i preparació del software necessari

NVM i Node.js

Per a poder compilar el codi font de l'aplicació és necessari tenir instal·lat **Node.js**, ja que el codi font de l'aplicació està escrit per aquesta *framework*. Per a poder gestionar de manera fàcil les versions instal·lades de **Node.js**, s'instal·larà l'eina **NVM** que permet migrar fàcilment entre les diferents versions de **Node.js**.

Per a poder instal·lar aquesta eina, és necessari descarregar un script que es proporciona al repositori oficial de **NVM** mitjançant la comanda:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
```

En cas, que l'script es trobi desactualitzat en el moment de la lectura d'aquest manual, es pot accedir al següent [enllaç](#) per a trobar la comanda corresponent amb l'URL de l'script actualitzat.

Un cop s'ha descarregat i executat l'script, ens apareixerà un avís el qual indica que cal reiniciar la consola (es surt de la consola amb la comanda **exit** i es torna a executar la consola o a iniciar sessió).

Quan es torna a posar en marxa la consola podem verificar que la instal·lació s'ha completat satisfactòriament introduint la comanda:

```
nvm --version
```

```
mmasser95@appSrv:~$ nvm --version
0.39.7
```

Si apareix un resultat com el de la captura anterior significa que la instal·lació del **NVM** s'ha realitzat correctament. A continuació, per a instal·lar l'última versió amb suport de llarg podem utilitzar una de les següents comandes:

```
nvm install 20
```

```
nvm install --lts
```

Un cop executada aquesta comanda ja estarà instal·lat a l'equip l'última versió **LTS** (*Long Time Service*) de **Node.js**. Podem verificar-ho mitjançant la comanda:

```
node --version
```

```
mmasser95@appSrv:~$ node --version
v20.11.1
```

Android Studio Command Tools

Quan es *minifica* l'aplicació amb Vite i posteriorment s'exporta l'aplicació a Android amb Capacitor, no es genera l'**APK** directament, sinó que genera un projecte d'Android Studio. Per a poder generar l'**APK** des del projecte generat, és necessari instal·lar aquesta eina. Per a fer-ho, caldrà anar a la [pàgina](#) de descàrregues d'Android Studio i anar a l'apartat "Solo herramientas de línea de comandos" i descarregar el fitxer de Linux, tal com es mostra en la captura següent:

Solo herramientas de línea de comandos

Plataforma	Paquete de herramientas del SDK	Tamaño	Suma de comprobación SHA-256
Windows	commandlinetools-win-11076708_latest.zip	153.6 MB	4d6931209eebb1bfb7c7e8b240a6a3cb3ab24479ea294f3539429574b1eec862
Mac	commandlinetools-mac-11076708_latest.zip	153.6 MB	7bc5c72ba0275c80a8f19684fb92793b83a6b5c94d4d179fc5988930282d7e64
Linux	commandlinetools-linux-11076708_latest.zip	153.6 MB	2d2d50857e4eb553af5a6dc3ad507a17adf43d115264b1afc116f95c92e5e258

Un cop descarregat el paquet i es troba disponible en el servidor caldrà descomprimir-lo mitjançant la comanda:

```
unzip commandlinetools-linux-11076708_latest.zip
```

Seguidament cal entrar a la carpeta descomprimida

```
cd commandlinetools-linux-11076708_latest/
```

I cal executar la comanda:

```
pwd
```

Un cop s'obté la ruta absoluta de la carpeta on es troba la instal·lació de les eines d'Android es copia i cal dirigir-se a editar el següent fitxer amb la comanda (el símbol "~" representa la ruta de la carpeta HOME de l'usuari que utilitza la consola):

```
nano ~/.bashrc
```

A continuació, cal afegir la línia següent al final del fitxer:

```
export ANDROID_HOME=<ruta_carpeta_commandlinetools>/Sdk
```

```
export ANDROID_HOME="/home/mmasser95/Android/Sdk"
```

Finalment, reiniciem la consola amb la comanda **exit** perquè s'efectuïn els canvis.

```
mmasser95@appSrv:~$ echo $ANDROID_HOME  
/home/mmasser95/Android/Sdk
```

Java JDK 17

Per a poder compilar el projecte d'Android Studio, l'eina anterior utilitza **Gradlew**. Una *framework* per a compilar codi Java en binaris **APK**. Per a poder funcionar correctament, **Gradlew** necessita com a mínim la versió 17 de Java JDK. A **Debian** per a poder instal·lar-lo podem utilitzar la següent comanda:

```
sudo apt install openjdk-17-jdk
```

A continuació, verifiquem que s'ha instal·lat correctament amb la comanda següent:

```
java --version
```

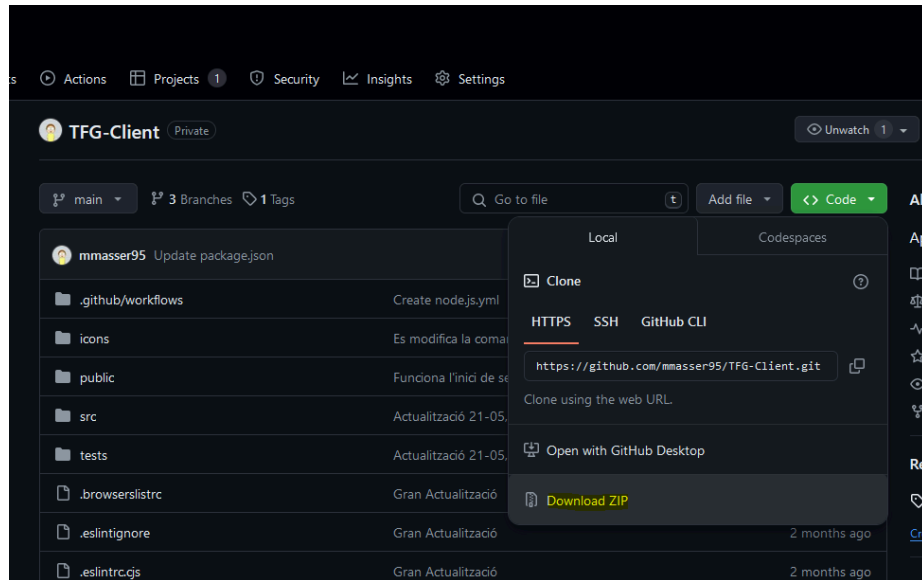
I per acabar, verifiquem que s'ha afegit la variable d'entorn **JAVA_HOME** correctament amb la comanda:

```
echo $JAVA_HOME
```

```
mmasser95@appSrv:~$ java --version
openjdk 17.0.10 2024-01-16
OpenJDK Runtime Environment (build 17.0.10+7-Debian-1deb12u1)
OpenJDK 64-Bit Server VM (build 17.0.10+7-Debian-1deb12u1, mixed mode, sharing)
mmasser95@appSrv:~$ echo $JAVA_HOME
/usr/lib/jvm/java-17-openjdk-amd64
```

Descàrrega del codi font i dependències

Per a poder descarregar el codi font de l'aplicació es pot descarregar des del [repositori oficial](#), tal com es mostra en la següent captura.



O clonant el repositori directament a la màquina mitjançant la comanda **git**. En cas de no tenir instal·lada aquesta eina es pot instal·lar amb la comanda:

```
sudo apt install git
```

Un cop instal·lada es pot descarregar l'última versió del codi font amb la comanda:

```
git clone https://github.com/mmasser95/TFG-Client.git
```

Després de descarregar el codi font, cal entrar a la carpeta descomprimida (descomprimir amb **unzip** en cas d'haver descarregat el projecte amb el navegador).

```
cd TFG-Client/
```

En aquesta carpeta, és on caldrà instal·lar les dependències necessàries per a compilar l'aplicació. Per a instal·lar-les es farà amb la comanda.

```
npm install
```


Claus i autoritzacions necessàries

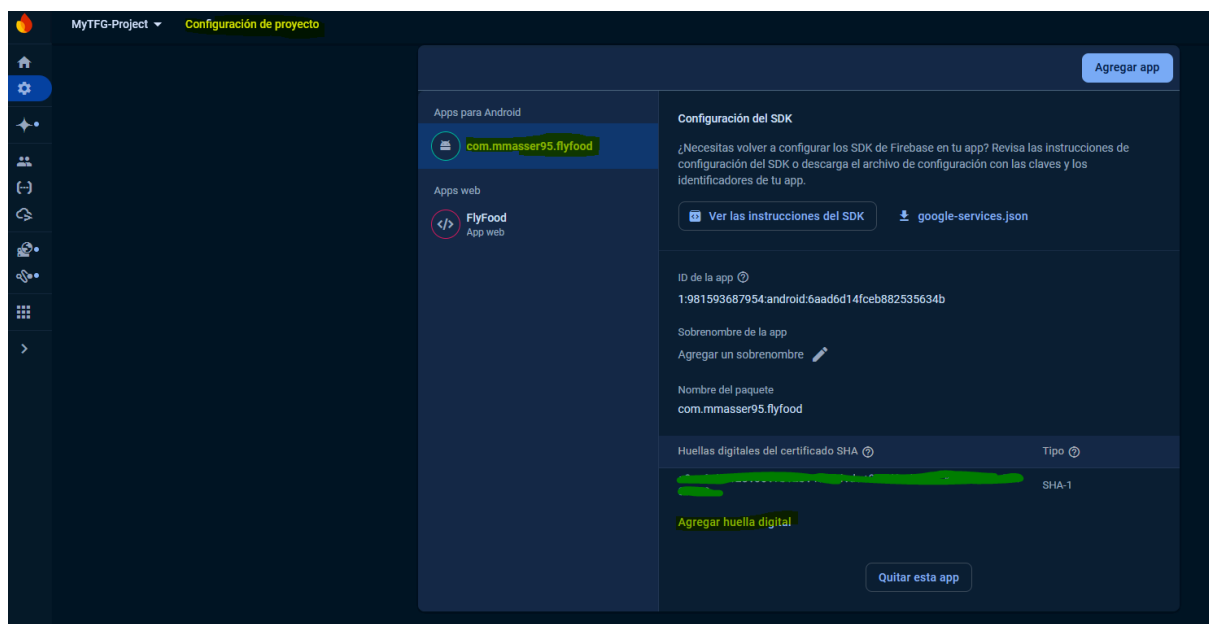
Perquè les notificacions i l'inici de sessió de Google funcionin correctament a l'aplicació compilada per Android és necessari afegir la signatura **SHA-1** al projecte de Firebase. Per a obtenir la signatura **SHA-1** és necessari executar la següent comanda:

```
keytool -list -v -keystore ~/.android/debug.keystore -alias androiddebugkey  
-storepass android -keypass android
```

Un cop executada ens apareixerà un resultat com el de la captura següent:

```
mmaser95@appSrv:~$ keytool -list -v -keystore ~/.android/debug.keystore -alias androiddebugkey -storepass android -keypass android  
Nombre de Alias: androiddebugkey  
Fecha de Creación: 4 mar 2024  
Tipo de Entrada: PrivateKeyEntry  
Longitud de la Cadena de Certificado: 1  
Certificado[1]:  
Propietario: C=US, O=Android, CN=Android Debug  
Emisor: C=US, O=Android, CN=Android Debug  
Número de serie: 1  
Válido desde: Mon Mar 04 22:25:27 CET 2024 hasta: Wed Feb 25 22:25:27 CET 2054  
Huellas digitales del certificado:  
SHA1: C0:A0:B3:28:6C:FB:2B:43:6D:DA:0A:D0:D4:39:04:7C:84:5C:0D:58  
SHA256: 8B:24:61:F7:8C:03:CF:3C:0E:F2:24:3D:E9:F0:AC:C0:9F:D1:55:01:DA:1B:B8:CF:D9:16:05:B2:5B:84:7C:DB  
Nombre del algoritmo de firma: SHA256withRSA  
Algoritmo de clave pública de asunto: Clave RSA de 2048 bits  
Versión: 1  
mmaser95@appSrv:~$
```

S'emmagatzema la signatura SHA1 i al projecte de Firebase s'afegeix aquesta firma a l'apartat següent:



Un cop afegida l'empremta digital al projecte s'haurà de descarregar el JSON amb informació d'autenticació de l'app i introduir-lo a la carpeta del projecte. També caldrà incloure l'AndroidManifest.xml inclòs en l'entrega. D'aquesta manera quan executem els scripts de compilació totes aquestes claus s'afegiran automàticament al projecte i l'aplicació utilitzarà correctament els serveis de Google. Com que el compte del projecte de Firebase es troba en un compte personal. *Per la qual cosa no es pot donar accés a aquest compte ni proporcionar credencials per a autoritzar compilacions.*

Compilació de l'aplicació

Per a poder facilitar la compilació de l'aplicació es faciliten scripts en el *package.json*, que permeten executar comandes llargues i difícils de recordar o d'escriure. S'han inclòs els següents scripts:

- **npm run build:** Aquesta comanda genera els fitxers necessaris per a compilar el projecte i generar els fitxers necessaris per a desplegar l'aplicació en un servidor web. Aquests fitxers els genera dins la carpeta */dist*. Addicionalment, exporta el projecte per a Android Studio. A continuació, s'afegeixen els fitxers de configuració necessaris per al projecte d'Android i finalment executa l'eina **Gradlew** per a generar l'**APK**. (Únicament utilitzar aquesta comanda la primera vegada que es compila l'aplicació)
- **npm run build2:** Aquesta comanda únicament s'ha d'utilitzar quan ja s'ha compilat una vegada l'aplicació o existeix la carpeta Android al projecte. Aquesta serveix per a actualitzar el projecte web i d'Android i generar l'aplicació **APK** quan s'han dut a terme canvis en el codi font. D'aquesta manera s'escurça el temps per a compilar nous canvis.

Web

Per a compilar l'aplicació per a versió web, caldrà utilitzar la comanda des de la carpeta arrel del projecte:

```
npm run build
```

Aquesta comanda genera una carpeta al projecte anomenada /dist, la qual conté l'estructura necessària per a ser desplegada en un servidor web. En aquest cas, s'explica com instal·lar l'aplicació amb nginx. Per a poder instal·lar nginx a l'equip caldrà executar la comanda:

```
sudo apt install nginx
```

A continuació, caldrà editar el fitxer de configuració de la següent manera:

```
sudo nano /etc/nginx/sites-default/default
```

```
GNU nano 7.2 /etc/nginx/sites-enabled/default *
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #

server_name app.flyfood.online api.flyfood.online app2.flyfood.online;

    location / {
        root /home/mmasserpis95/TFG-Client/dist;
        index index.html;
        try_files $uri $uri/ /index.html;
    }

    location /api {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        #try_files $uri $uri/ =404;
        proxy_pass http://localhost:5000/api;
    }

    location /uploads {
        proxy_pass http://localhost:5000/uploads;
    }
}
```

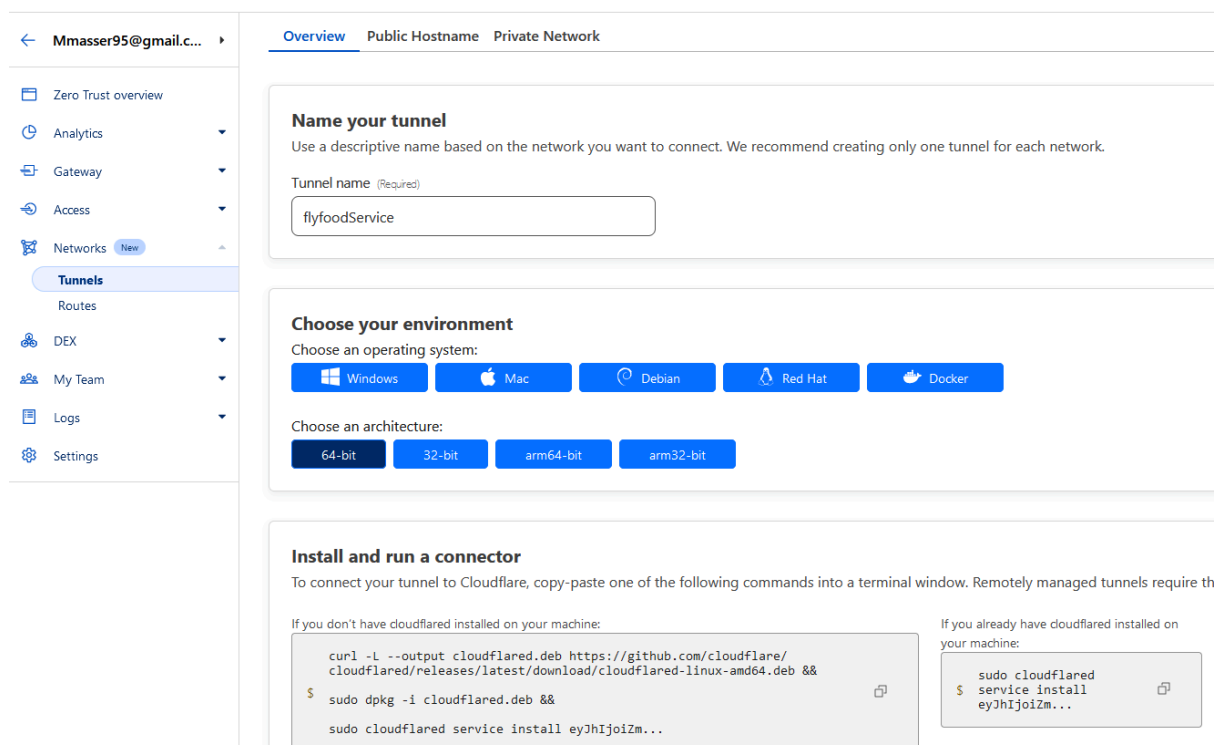
En la configuració al paràmetre root afegim la ruta on es troba la carpeta /dist generada anteriorment. Al paràmetre server_name, afegim el nom del domini el qual ha de respondre el servidor web. Finalment, per aconseguir que les rutes de l'app funcionin correctament en els navegadors, cal afegir un proxy_pass el qual reencamina les rutes /login i les executa a través de l'arxiu index.html. D'aquesta manera en refrescar l'app al navegador, no apareixerà un error 404.

Tunneling amb Cloudflare

Si no es desitja utilitzar Cloudflare perquè és necessari tenir en propietat un nom de domini. Es pot instal·lar un certificat mitjançant **certBot** de **Let's Encrypt** i utilitzar un domini o subdomini gratuït. Les instruccions es poden trobar en el següent [enllaç](#).

Per a poder configurar de forma còmoda un certificat al servei web i afegir seguretat extra a l'aplicació es pot "tunelitzar" el servidor a través de **Cloudflare**. Per a fer-ho caldrà registrar-nos a **Cloudflare**. Registrar un domini i anar a la consola **Zero Trust**. Un cop aquí caldrà afegir un nou túnel, seguint les instruccions que ens demana.

Quan el túnel estigui configurat a la consola **Zero Trust**, a la màquina caldrà instal·lar i iniciar el túnel mitjançant la primera comanda o si tenim ja està **Cloudflare Tunnels**, caldrà executar la segona comanda per inicialitzar únicament el servei en aquesta màquina.



The screenshot shows the Cloudflare Zero Trust console interface. The left sidebar contains navigation options: Zero Trust overview, Analytics, Gateway, Access, Networks (with a 'New' badge), Tunnels (selected), Routes, DEX, My Team, Logs, and Settings. The main content area is titled 'Overview' and has tabs for 'Public Hostname' and 'Private Network'. It is divided into three sections:

- Name your tunnel:** A form with a 'Tunnel name (Required)' field containing 'flyfoodService'.
- Choose your environment:** Options for 'Choose an operating system' (Windows, Mac, Debian, Red Hat, Docker) and 'Choose an architecture' (64-bit, 32-bit, arm64-bit, arm32-bit).
- Install and run a connector:** Instructions to copy-paste commands into a terminal. It provides two sets of commands: one for machines without Cloudflare Tunnel installed and one for machines with it already installed.

```
curl -L --output cloudflared.deb https://github.com/cloudflare/cloudflared/releases/latest/download/cloudflared-linux-amd64.deb &&
sudo dpkg -i cloudflared.deb &&
sudo cloudflared service install eyJhIjoizM...
```

```
sudo cloudflared
service install
eyJhIjoizM...
```

Un cop iniciat el túnel, l'aplicació es trobarà publicada al domini especificat sense haver d'obrir els ports mitjançant la connexió segura dels túnels de **Cloudflare**. A més a més, la pàgina gaudirà de les característiques de seguretat que proporciona el **DNS** de **Cloudflare**.

Android

Per a compilar l'aplicació per primera vegada per a la plataforma Android, cal executar la comanda a la carpeta arrel del projecte:

```
npm run build
```

Aquesta comanda pot tardar una bona estona en acabar la seva execució, ja que primer compila la versió web i a continuació efectua totes les operacions necessàries per a generar l'**APK**. (La seva execució pot durar fins a 15 minuts, segons les especificacions de la màquina on es compila).

Si s'han realitzat correctament cada un dels passos especificats en aquest manual, un cop acabada l'execució de la comanda, s'haurà d'haver generat un fitxer **APK** a la ruta:

```
{carpeta_projecte}/app/build/outputs/apk/debug/app-debug.apk
```

Un cop generada l'**APK** es pot enviar a un dispositiu mòbil per a poder instal·lar-la i provar-la. La instal·lació de l'**APK** en Android i les instruccions d'ús de l'aplicació es troben al fitxer *Manual d'ús de l'aplicació*.