Asteroides (Click)

Documentação Build/Deploy (Versão 1)

Este documento consolida o fluxo completo para: GitHub Pages (HTTPS) + PWA (manifest/ícones), geração de app Android via Bubblewrap (TWA), associação Digital Asset Links (assetlinks.json), build e assinatura (APK/AAB), e atualização por versionamento (update correto).

Gerado em 26/02/2026

Sumário

1. Pré-requisitos e dependências

1.0 Aviso crítico: VPN

Desligue a VPN antes de init/build do Bubblewrap, downloads de JDK/SDK e builds Gradle. Durante o projeto, erros de rede (ex.: ECONNRESET) ocorreram quando a VPN estava ativa.

1.1 Downloads oficiais

- Node.js LTS: https://nodejs.org/en/download
- Microsoft Build of OpenJDK 17: https://learn.microsoft.com/en-us/java/openjdk/download
- Android Studio: https://developer.android.com/studio
- sdkmanager (referência): https://developer.android.com/tools/sdkmanager
- Bubblewrap CLI (npm): https://www.npmjs.com/package/@bubblewrap/cli

1.2 Instalação via PowerShell (winget)

```
winget install -e --id OpenJS.NodeJS.LTS
winget install -e --id Microsoft.OpenJDK.17
winget install -e --id Google.AndroidStudio
```

1.3 Verificações rápidas

```
node -v
npm -v
java -version
```

1.4 Caminhos típicos (Windows)

Item	Caminho típico
Android SDK	%LOCALAPPDATA%\Android\Sdk
adb.exe	%LOCALAPPDATA%\Android\Sdk\platform-tools\adb.exe
apksigner / zipalign	%LOCALAPPDATA%\Android\Sdk\build-tools\<versão>\apksigner.bat e zipalign.exe
sdkmanager	%LOCALAPPDATA%\Android\Sdk\cmdline-tools\latest\bin\sdkmanager.bat

2. Deploy web no GitHub Pages (PWA)

2.1 Estrutura mínima publicada

• index.html
• styles.css
• manifest.webmanifest
• icons/icon-192.png
• icons/icon-512.png

2.2 Checklist rápido (sem 404)

```
https://ruanbonina.github.io/aster-click/
https://ruanbonina.github.io/aster-click/manifest.webmanifest
https://ruanbonina.github.io/aster-click/icons/icon-192.png
https://ruanbonina.github.io/aster-click/icons/icon-512.png
```

2.3 Referência do manifest no HTML

```
<link rel="manifest" href="./manifest.webmanifest">
```

3. TWA (Bubblewrap): init e configuração

3.1 Criar pasta e inicializar a partir do Web Manifest

```
mkdir C:\dev\aster-click-twa
cd C:\dev\aster-click-twa

npx @bubblewrap/cli init --manifest=https://ruanbonina.github.io/aster-click/manif
est.webmanifest
```

3.2 Informações do init (exemplo)

Domain: ruanbonina.github.io
URL path: /aster-click/
Icon URL: https://ruanbonina.github.io/aster-click/icons/icon-512.png
Application ID (package): io.github.ruanbonina.asterclick
Display mode: fullscreen

4. Digital Asset Links (assetlinks.json) e verificação

4.1 O que é e por que importa

Para o app abrir como TWA (sem barra de navegador), o Android valida o site via assetlinks.json usando o SHA-256 do certificado do app. Se a validação falhar, o app pode cair para modo navegador.

4.2 Ponto crítico (GitHub Pages)

No GitHub Pages, foi necessário publicar o diretório **sem ponto** no começo: well-known em vez de .well-known.

4.3 URL final (deve abrir sem 404)

```
https://ruanbonina.github.io/aster-click/well-known/assetlinks.json
```

4.4 Exemplo de conteúdo

```
[
 {
 "relation": ["delegate_permission/common.handle_all_urls"],
 "target": {
 "namespace": "android_app",
 "package_name": "io.github.ruanbonina.asterclick",
 "sha256_cert_fingerprints": [
 "0A:62:23:AF:46:70:06:B3:5E:E9:AC:F2:73:C5:7C:1D:F7:4D:1C:A6:52:E7:28:AB:02:E5:3A:
97:A2:F5:18:56"
 ]
 }
 }
]
```

5. Build (APK/AAB)

5.1 Build via Bubblewrap

```
npx @bubblewrap/cli build
```

5.2 Build via Gradle (alternativa)

```
.\gradlew.bat assembleRelease
.\gradlew.bat bundleRelease
```

6. Assinatura e verificação do APK

6.1 Assinar manualmente (zipalign + apksigner)

```
$bt = "$env:LOCALAPPDATA\Android\Sdk\build-tools\36.0.0"

& "$bt\zipalign.exe" -p 4 `
 ".\app\build\outputs\apk\release\app-release-unsigned.apk" `
 ".\app\build\outputs\apk\release\app-release-aligned.apk"

& "$bt\apksigner.bat" sign `
 --ks "$env:USERPROFILE\android.keystore" `
 --ks-key-alias android `
 --out ".\app\build\outputs\apk\release\app-release-signed.apk" `
 ".\app\build\outputs\apk\release\app-release-aligned.apk"
```

6.2 Verificar assinatura

```
$bt = "$env:LOCALAPPDATA\Android\Sdk\build-tools\36.0.0"
& "$bt\apksigner.bat" verify --verbose `
 ".\app\build\outputs\apk\release\app-release-signed.apk"
```

Observação: quando o apksigner pedir senha, ela não aparece na tela. Digite normalmente e pressione Enter.

7. Instalação no device (adb)

7.1 Listar devices e instalar/atualizar

```
$adb = "$env:LOCALAPPDATA\Android\Sdk\platform-tools\adb.exe"
& $adb devices

# instalar/atualizar mantendo dados
& $adb install -r ".\app\build\outputs\apk\release\app-release-signed.apk"
```

7.2 Se o device não aparece

- Confirme Depuração USB habilitada e aceite o prompt no aparelho.
- Se houver bloqueio do Windows/antivírus, libere o ADB e replugue o cabo.
- Troque cabo/porta USB se necessário.

8. Versionamento e update correto (TWA)

8.1 Regra de ouro

appVersionCode sempre precisa ser incrementado a cada release (1 → 2 → 3...). Sem isso, o Android não considera update e pode falhar na instalação.

8.2 Onde versionar

O projeto TWA mantém versões em twa-manifest.json (e/ou Gradle dependendo do template). Use appVersionName para versão legível e appVersionCode como inteiro incremental.

8.3 Passo a passo (PowerShell) pronto para copiar e colar

Cole este bloco no PowerShell dentro da pasta do projeto TWA (onde existe twa-manifest.json). Ele atualiza versão, executa build e instala como update via adb.

```
# (0) Desligue a VPN antes de começar.

# (1) Vá para a pasta do projeto TWA
cd C:\dev\aster-click-twa

# (2) Atualize versões no twa-manifest.json (edita JSON automaticamente)
$manifestPath = ".\twa-manifest.json"
$j = Get-Content $manifestPath -Raw | ConvertFrom-Json
$j.appVersionName = "1.0.1" # <-- troque aqui (ex.: 1.0.2)
$j.appVersionCode = 2 # <-- SEMPRE incremente (1 -> 2 -> 3...)
$j | ConvertTo-Json -Depth 50 | Set-Content $manifestPath -Encoding UTF8

# (3) Build
npx @bubblewrap/cli build

# (4) (Se necessário) Assinar manualmente o APK de release
$bt = "$env:LOCALAPPDATA\Android\Sdk\build-tools\36.0.0"

& "$bt\zipalign.exe" -p 4 `
 ".\app\build\outputs\apk\release\app-release-unsigned.apk" `
 ".\app\build\outputs\apk\release\app-release-aligned.apk"

& "$bt\apksigner.bat" sign `
 --ks "$env:USERPROFILE\android.keystore" `
 --ks-key-alias android `
 --out ".\app\build\outputs\apk\release\app-release-signed.apk" `
 ".\app\build\outputs\apk\release\app-release-aligned.apk"

# (5) Update no device (USB depuração ativa)
$adb = "$env:LOCALAPPDATA\Android\Sdk\platform-tools\adb.exe"
& $adb devices
& $adb install -r ".\app\build\outputs\apk\release\app-release-signed.apk"
```

9. Troubleshooting rápido

Sintoma	Causa provável	Ação direta
App abre como navegador (barras de URL).	assetlinks.json inacessível/404 ou escopo errado	Garanta URL sem 404 e reinstale o app após public
keytool não reconhecido	JAVA_HOME/PATH não aponta para JDK	Use caminho completo: C:\Program Files\Java\jdk-1
apksigner pede senha e parece travar	Senha não aparece no console	Digite normalmente e pressione Enter.
adb não lista device	Depuração USB não autorizada	Rode: %LOCALAPPDATA%\Android\Sdk\platform-t
Erros de rede (ECONNRESET) durante init/build	VPN/inspeção de rede	Desligue a VPN e tente novamente.