Manuale Operativo – MedReportAndSign

# 1. Prerequisiti

- Node.js e npm installati.  
- Repository GitHub già configurato.  
- Permessi di push/tag sul repository.  
- Account GitHub collegato.  
- Tutte le dipendenze installate (npm install).

Accedi al terminale come Amministratore

# 2. Aggiornamento Versione

Aggiorna la versione in package.json:

"version": "x.y.z" (oppure npm version x.y.z)

# 3. Compilazione del Progetto

Esegui in sequenza:

npm run build  
npx electron-builder --win

# 4. Versionamento Git

git checkout main  
git pull origin main  
git add .  
git commit -m "Messaggio descrittivo"  
git push origin main

Se hai aggiornato solo la versione:

git add package.json  
git commit -m "Aggiorna versione a x.y.z"  
git push origin main

# 5. Tag della Release

git tag -a vx.y.z -m "Release vx.y.z: descrizione"  
git push origin vx.y.z

# 6. Build Pacchetti Installer

Per Windows:

npx electron-builder --win

Troverai i file in /dist:  
- MedReportAndSign-Setup-x.y.z.exe  
- latest.yml (per aggiornamento automatico)

# 7. Pubblicazione Release su GitHub

A. Caricamento Manuale:

1. Vai su GitHub → Releases → Draft new release  
2. Seleziona il tag appena creato (o creane uno nuovo).  
3. Inserisci una descrizione della release.  
4. Carica i file binari generati dalla build (.exe, .yml, eventuali altri).  
5. Clicca su "Publish release".

B. Caricamento Automatico:

npx electron-builder --publish always

Assicurati di avere un token GitHub (GH\_TOKEN) valido come variabile d’ambiente.

# 8. Personalizzazione Nome Installer

Definisci il nome dell’exe in package.json sotto build → win → artifactName:

"artifactName": "MedReportAndSign-Setup-${version}.exe"

# 9. Best Practice

- Esegui sempre git pull prima di lavorare.  
- Aggiorna il numero di versione per ogni release.  
- Tagga sempre la release dopo il commit finale.  
- Non caricare file temporanei o personali su Git.  
- Dopo la pubblicazione, testa sempre l’installer scaricandolo da GitHub.

# 10. Riepilogo Procedura Completa

# (1) Aggiorna versione  
# (2) Compila progetto  
npm run build  
  
# (3) Versiona e pubblica su Git  
git checkout main  
git pull origin main  
git add .  
git commit -m "Messaggio release vx.y.z"  
git push origin main  
  
# (4) Tag  
git tag -a vx.y.z -m "Release vx.y.z: descrizione"  
git push origin vx.y.z  
  
# (5) Builda installer  
npx electron-builder --win  
  
# (6) Carica su GitHub Release i file da /dist  
# oppure pubblica automaticamente:  
npx electron-builder --publish always

# Note Finali

- Puoi aggiungere istruzioni o screen specifici.  
- Se usi anche macOS o Linux, lancia:  
 npx electron-builder --mac  
 npx electron-builder --linux  
 (verranno generati .dmg e .AppImage/.deb in dist).

Redatto da: Dharma Healthcare

Data: [inserisci data rilascio]