Procedimento — Atualizar o App (Backup + Publicação Segura)

Este guia explica como atualizar o aplicativo com segurança: criar backup (tag), publicar no GitHub e recompilar na Streamlit Cloud.

# 1) Onde está o projeto no seu PC

* Projeto local (padrão): C:\Users\jcano\Documents\Sistema Compras
* Arquivo principal: reposicao\_facil.py
* Dependências: requirements.txt
* Uploads persistidos: .uploads\ALIVVIA\ e .uploads\JCA\ (salvos pelo app)

# 2) Preparar backup (tag) antes de mudar

Crie uma tag de backup para poder voltar se algo der errado.

* Abrir PowerShell na pasta do projeto
* Executar: git tag -a vX.Y.Z-backup -m backup antes de atualizar
* Executar: git push origin vX.Y.Z-backup

# 3) Publicar mudanças no GitHub

* Salvar o arquivo editado (ex.: reposicao\_facil.py)
* git checkout main
* git pull origin main
* git add -A
* git commit -m "release: atualização do app"
* git push origin main

# 4) Recompilar na Streamlit Cloud

* Abrir o app na Streamlit Cloud
* Rodapé → Manage app
* Settings → Advanced → Clear cache
* Clicar em Reboot
* Atualizar a aba do app (F5)

# 5) Teste curto (garantia)

* Aba “Dados das Empresas”: verificar selo “salvo” nos 3 arquivos de cada empresa
* Aba “Compra Automática”: gerar cálculo padrão (H/G/LT simples)
* Aplicar 1 filtro leve (fornecedor/SKU)
* Exportar XLSX e conferir se não está vazio

# 6) Rollback (voltar versão)

Se a nova versão der problema, volte para a versão estável.

* git fetch --all --tags
* git reset --hard v3.3.1-ok (tag estável conhecida)
* git push --force origin main

# 7) Onde salvar os arquivos gerados

* Saídas: C:\Alivvia\Reposicao\outputs\ALIVVIA\pedido\_ALIVVIA\_{YYYYMMDD}.xlsx
* Saídas: C:\Alivvia\Reposicao\outputs\JCA\pedido\_JCA\_{YYYYMMDD}.xlsx
* Comparativos: C:\Alivvia\Reposicao\outputs\Comparativos\Compra\_Duas\_Contas\_{YYYYMMDD}.xlsx