AgroSolutions — Guia Local (Docker Desktop Kubernetes)
Execução dos 4 Microsserviços em Containers + Kubernetes

Versão: pacote de correções (gerado em 22/01/2026).

Pré-requisitos

- Windows 10/11 com Docker Desktop instalado e em execução.
- Kubernetes habilitado no Docker Desktop (Settings → Kubernetes → Enable Kubernetes).
- kubectl instalado e apontando para o contexto do Docker Desktop (geralmente `docker-desktop`).
- PowerShell (recomendado) e Git (opcional, mas útil).

1) Verificar o cluster local

No PowerShell, valide o contexto e o acesso ao cluster:

```
kubectl config get-contexts
kubectl config current-context
kubectl get nodes
```

2) Build das imagens Docker (tag local)

No diretório raiz do repositório, execute:

```
.\build\scripts\docker-build.ps1 -Environment local -Registry ghcr.io/agrosolutions
```

Valide se as quatro imagens foram criadas com a tag `local`:

```
docker images | findstr ghcr.io/agrosolutions
```

3) Subir o ambiente no Kubernetes (overlay local)

Aplicar os manifests via Kustomize (kubectl -k):

```
kubectl apply -k .\infra\k8s\overlays\local
```

Isso criará o namespace `agrosolutions-local` e aplicará ConfigMaps, Deployments e Services dos 4 microsserviços.

4) Validar se tudo ficou "Ready"

```
kubectl get ns
kubectl get pods -n agrosolutions-local
kubectl get svc  -n agrosolutions-local
```

Se algum Pod não estiver Ready, veja eventos e logs:

```
kubectl describe pod <POD_NAME> -n agrosolutions-local
kubectl logs <POD_NAME> -n agrosolutions-local
```

5) Acessar Swagger/API localmente (port-forward)

Em quatro terminais (ou em sequência), execute:

```
kubectl port-forward -n agrosolutions-local svc/usuarios      8081:80
kubectl port-forward -n agrosolutions-local svc/propriedades  8082:80
kubectl port-forward -n agrosolutions-local svc/ingestao      8083:80
kubectl port-forward -n agrosolutions-local svc/analise       8084:80
```

Acesse no navegador:

- http://localhost:8081/swagger (Usuarios)
- http://localhost:8082/swagger (Propriedades)
- http://localhost:8083/swagger (Ingestao)
- http://localhost:8084/swagger (Analise)

6) Derrubar o ambiente local

Remover recursos aplicados pelo overlay:

```
kubectl delete -k .\infra\k8s\overlays\local
```

Alternativa (remove tudo do namespace):

```
kubectl delete namespace agrosolutions-local
```

Troubleshooting rápido

A) Pods em ImagePullBackOff / ErrImagePull

Causas comuns:

- As imagens não existem com a tag esperada (ex.: overlay usa `local`, mas você buildou com SHA).
- Imagem aponta para registry diferente do build.
- Você está usando um cluster que não enxerga imagens locais (não é o caso do Docker Desktop Kubernetes; aqui normalmente enxerga).

Verifique quais imagens o deployment está usando:

```
kubectl get deploy -n agrosolutions-local -o wide
kubectl describe deploy <DEPLOYMENT> -n agrosolutions-local | findstr Image
```

B) Pods reiniciando (Readiness/Liveness falhando)

Verifique se os endpoints de health estão respondendo dentro do Pod:

```
kubectl exec -n agrosolutions-local -it <POD_NAME> -- sh
# dentro do pod:
wget -qO- http://localhost:8080/health/live
wget -qO- http://localhost:8080/health/ready
```

C) Swagger não aparece

O Swagger está habilitado apenas em Development. No ambiente local, o overlay define `ASPNETCORE_ENVIRONMENT=Development` via patch em ConfigMap.

Confirme a variável no Pod:

```
kubectl exec -n agrosolutions-local -it <POD_NAME> -- printenv | findstr ASPNETCORE_ENVIRONM
```