

DATA SHEET · MIGRAÇÃO

Migração GitLab™ para GitLab self-managed

Grupos e projetos de GitLab.com ou de GitLab self-managed migrados em ondas para a instância GitLab self-managed do cliente, por Direct Transfer orquestrado pelo GitLab Congregate

CÓDIGO**PS-MIG-01****TIPO****Data sheet****VERSÃO****1.2****DATA****1 de outubro de 2026**

O3S Consultoria e Tecnologia da Informação Ltda.
contato@pointertech.digital · +55 (61) 99874-4648 · pointertech.digital

GITLAB is a trademark of GitLab Inc. in the United States and other countries and regions.

**Select
Partner**

DATA SHEET · PS-MIG-01

Migração GitLab para GitLab self-managed

Grupos e projetos de GitLab.com ou de GitLab self-managed migrados em ondas para a instância GitLab self-managed do cliente, por Direct Transfer orquestrado pelo GitLab Congregate

Migração de grupos e projetos de GitLab.com ou de uma instância GitLab self-managed para a instância GitLab self-managed do cliente, inclusive uma instância implantada pela Pointer. O método padrão é o **Direct Transfer**, recurso nativo da plataforma GitLab, orquestrado pelo **GitLab Congregate** e disparado por onda pelo pipeline Pointer, que também executa o pós-importação, a verificação no destino, a contagem independente origem × destino e o rollback da onda. Quando o Direct Transfer não é possível (rede ou versão), a alternativa é o método por arquivo (exportação e importação de projetos).

1. Para quem

- Grupos e projetos em GitLab.com que passam a operar em uma instância GitLab self-managed do cliente.
- Instância GitLab self-managed que será substituída ou consolidada em outra instância GitLab self-managed, inclusive uma instância implantada pela Pointer.
- Instância de origem sem conectividade HTTPS a partir do destino ou fora da regra de versões do Direct Transfer: método por arquivo.
- Migrações que precisam associar autoria e memberships aos usuários do destino, criados na migração ou já provisionados por LDAP ou SAML.

2. Onde executamos

O host de migração fica na rede do cliente (data center próprio ou nuvem), só com conexões de saída para a origem, o destino e os serviços da GitLab usados pelo pipeline. A instância GitLab de destino é a do cliente, on-premises ou em nuvem. O Direct Transfer exige que a instância de destino alcance a origem por HTTPS.

3. Escopo do serviço

- Discovery de migração e revisão do inventário da origem (Levantamento da Plataforma, quando contratado, e inventário gerado pelo pipeline).
- Plano de ondas: grupos inteiros e projetos avulsos por onda, também por planilha, com destino de cada item, janela, aprovador e janela de rollback.

- Preparação do projeto do engajamento e da stack do GitLab Congregate no host de migração do cliente, com verificação prévia (preflight) contra a origem e o destino.
- Execução das ondas por Direct Transfer (ou pelo método por arquivo): onda piloto, simulação, migração com aprovação, pós-importação, verificação no destino e contagem independente origem x destino.
- Usuários: criação pelo GitLab Congregate ou uso de usuários já provisionados no destino, e reatribuição em lote dos usuários placeholder.
- Cópia das imagens de contêiner da origem para o registry de destino, quando incluída no plano.
- Reescrita de referências à origem nos repositórios (relatório ou um merge request por projeto).
- Corte por onda: arquivamento dos projetos na origem, com opção de desfazer.
- Rollback da onda dentro da janela acordada e remigração no mesmo caminho.
- Encerramento: remoção dos dados do engajamento no host de migração, remoção do runner, revogação dos tokens e relatório de encerramento.

4. Origens, destino e métodos suportados

- **Disponível** Ofertado nas condições deste documento.

Origem	Destino	Método	Versões mínimas	Estado
GitLab.com	GitLab self-managed	Direct Transfer orquestrado pelo GitLab Congregate	Destino 16.8 ou superior; origem no máximo duas versões minor anteriores ao destino	• Disponível
GitLab self-managed	GitLab self-managed	Direct Transfer orquestrado pelo GitLab Congregate	Origem e destino 16.8 ou superior; origem no máximo duas versões minor anteriores ao destino	• Disponível
GitLab self-managed	GitLab self-managed	Exportação e importação por arquivo orquestradas pelo GitLab Congregate (alternativa ao Direct Transfer)	Destino na mesma versão da origem ou posterior; exportação de até duas versões minor anteriores ao destino	• Disponível

- Destino: somente GitLab self-managed, inclusive instância implantada pela Pointer. GitLab.com e GitLab Dedicated como destino estão no roadmap.
- Regra de versões do Direct Transfer (documentação GitLab): origem e destino na versão 16.8 ou superior e origem no máximo duas versões minor anteriores ao destino. O preflight reprova fora da regra e gera alerta quando a origem é mais nova que o destino.
- Método por arquivo: exige token de administrador na origem e no destino (pipeline Pointer).
- Recursos pagos (por exemplo épicos, regras de aprovação e push rules) só migram com o nível de assinatura correspondente no destino.

- Outras origens: GitHub (PS-MIG-02), Azure DevOps (PS-MIG-03) e Bitbucket Cloud (PS-MIG-04). Bitbucket Server/Data Center e AWS CodeCommit estão no roadmap.

5. Recursos configurados

- Nativo — recurso da plataforma GitLab
 - Requer configuração
- Integração externa — depende de serviço do cliente ou de terceiro

Recurso	Classificação	Detalhe
Direct Transfer	• Nativo	Recurso nativo da plataforma GitLab: a instância de destino busca grupos e projetos na origem pela API. Disparado pelo GitLab Congregate a partir do pipeline, onda a onda.
Ondas de grupos e de projetos avulsos, também por planilha	• Requer configuração	Cada onda reúne grupos inteiros (com subgrupos e projetos) e projetos avulsos; a planilha tem as colunas onda e grupo ou projeto, e o inventário da origem serve de modelo. Destino de cada item = grupo pai no destino + caminho na origem. Onda com grupos e projetos avulsos roda em dois ciclos: grupos, depois projetos.
Simulação da onda	• Requer configuração	Execução do GitLab Congregate em modo simulação, sem criar nada no destino; no Direct Transfer, confere cada item do pacote simulado contra o plano da onda e reprova caminho de destino já ocupado. A migração da onda só fica disponível depois da simulação com sucesso.
Pós-importação	• Requer configuração	O GitLab Congregate recria itens que o Direct Transfer não migra, como variáveis de CI/CD, ambientes, feature flags, deploy keys, webhooks, regras de aprovação, pacotes e Terraform states (detalhe na matriz do documento de pré-requisitos). Itens pagos só são recriados com o nível de assinatura da origem informado no plano.
Cópia das imagens de contêiner	• Requer configuração	Opcional. O pipeline copia as imagens tag a tag com o Docker do host de migração (login temporário, pull, tag, push e remoção local).
Criação de usuários	• Requer configuração	Opcional. O GitLab Congregate cria no destino os usuários da origem, com token de administrador nos dois lados; antes das ondas, no método por arquivo, ou junto com cada onda, no Direct Transfer.

Recurso	Classificação	Detalhe
Reatribuição em lote dos usuários placeholder	• Requer configuração	No Direct Transfer, contribuições e memberships chegam em usuários placeholder, mecanismo nativo da plataforma GitLab, mesmo quando o usuário já existe no destino. O pipeline reatribui em lote pela API de reatribuição (correspondência por e-mail, username ou planilha origem → destino), acompanha o estado e reaplica os aprovadores das regras de aprovação. Sem a opção de reatribuição sem confirmação no destino, cada usuário aceita o pedido.
Verificação no destino e contagem independente origem x destino	• Requer configuração	Por API: existência de cada item, número de projetos por grupo e status do Direct Transfer. Contagem item a item: branches, tags, issues, merge requests e subgrupos a menos no destino reprovam a onda; releases, milestones, variáveis, ambientes, agendamentos, webhooks, feature flags, pacotes, labels, push rule, regras de aprovação, membros, tags de registry, bytes de LFS, épicos e autoria em placeholder divergentes geram alerta. Inclui o relatório de diferenças do GitLab Congregate.
Rollback da onda na janela	• Requer configuração	Remove do destino, em ordem inversa, os grupos e projetos da onda criados dentro da janela (padrão 24 horas, de 1 a 720 horas); exclusão agendada por padrão ou permanente. Com cópia de imagens, remove antes as imagens. Remove o grupo intermediário vazio criado na janela. Usuários não são removidos.
Remigração	• Requer configuração	O rollback com exclusão permanente libera o caminho no destino para migrar a mesma onda de novo.
Reescrita de referências à origem	• Requer configuração	O GitLab Congregate procura nos repositórios da onda URLs, remotos SSH e include de CI/CD que apontam para a origem. Por padrão só gera relatório; com aplicação, abre um merge request por projeto para revisão do cliente.
Arquivamento da origem (corte)	• Requer configuração	Arquiva na origem os projetos da onda, que ficam somente leitura, e registra os que já estavam arquivados; o desfazer não desarquiva o que o cliente já tinha arquivado.

Recurso	Classificação	Detalhe
Tratamento de segredos no host	• Requer configuração	A configuração do GitLab Congregate com os tokens é gerada no início de cada job e apagada no fim; toda saída passa por um sanitizador que remove os tokens e as formas base64 deles e mascara e-mails (ficam a primeira letra e o domínio). Os logs sem sanitização ficam só no host de migração.

6. Ferramentas

FERRAMENTA DA GITLAB PROFESSIONAL SERVICES (LICENÇA MIT)

GitLab Congregate

Versão 8.6.0 fixada por digest, com MongoDB 8.0.12 e Redis 6.2 também fixados, em uma stack de contêineres exclusiva do engajamento no host de migração, sem portas publicadas. Lista a origem, encena os grupos e projetos de cada onda, dispara o Direct Transfer ou a exportação e importação por arquivo, executa o pós-importação, a reescrita de referências e o rollback.

RECURSO NATIVO DA PLATAFORMA GITLAB

Direct Transfer

Método padrão: migra grupos e projetos pela API, executado pela instância de destino.

RECURSO NATIVO DA PLATAFORMA GITLAB

Exportação e importação de projetos por arquivo

Método alternativo quando o Direct Transfer não é possível (rede ou versão).

FERRAMENTA POINTER

psctl

Valida o plano de migração, gera e apaga a configuração do GitLab Congregate em cada job, executa o preflight por API, resolve cada onda, verifica o destino, faz a contagem independente, reatribui os usuários placeholder, sanitiza as saídas e gera os relatórios.

COMPONENTE OFICIAL DA GITLAB

GitLab Runner

Runner de executor shell, exclusivo do projeto do engajamento, instalado no host de migração do cliente; executa todos os jobs exceto os de validação.

7. Como o pipeline Pointer executa



Etapas sob demanda: Criação dos usuários antes das ondas (**usuarios**); Referências à origem nos repositórios (**referencias**); Corte: origem somente leitura (**arquivar-origem**); Desfazer o corte (**desarquivar-origem**); Reversão da onda (**rollback**) (**rollback**).

- Validação do plano de migração** (automática a cada alteração): origem, destino, host, usuários, ondas e variáveis exigidas.
- Preparação do host de migração**: stack do GitLab Congregate com versões fixas, sem portas publicadas.
- Verificação prévia da migração** (automática): tokens, versões, Direct Transfer habilitado, grupo pai no destino, papel Owner na origem e configuração do GitLab Congregate.
- Inventário da origem**: grupos, projetos e usuários do escopo, tamanhos de repositório, LFS, pacotes e registry, projetos fora de ondas.
- Criação dos usuários antes das ondas** (opcional; método por arquivo).
- Por onda: **Simulação da onda** → **Migração da onda** (com aprovação), com pós-importação, cópia de imagens, reatribuição, verificação no destino e contagem origem x destino.
- Depois da onda: **Referências à origem nos repositórios**, **Corte: origem somente leitura** (e **Desfazer o corte**) e **Reversão da onda (rollback)** dentro da janela.
- Encerramento: remoção dos dados do host**.

8. Metodologia

Assessment → Implantação → Otimização → Transferência de conhecimento.

Fase 1**Assessment**

- Kick-off: escopo (origem, destino, volumes), janelas, congelamento de escrita na origem por onda, aprovadores de cada onda e do rollback.
- Discovery de migração: versões e método, nível de assinatura da origem, rede destino → origem, usuários e mapeamento, registries e pacotes, CI/CD e integrações; respostas registradas no registro de decisões do engajamento.
- Envio do documento de pré-requisitos (PR-MIG-01) e acompanhamento do checklist de prontidão.
- Criação do projeto do engajamento e validação do plano de migração pelo pipeline.
- Preparação do host de migração, verificação prévia (preflight) sem erro e inventário da origem.
- Plano de ondas: onda piloto e ondas seguintes, com nenhum projeto fora de onda sem decisão registrada.

Fase 2**Implantação**

- Criação dos usuários antes das ondas, quando o método por arquivo ou a criação de usuários estiver no plano.
- Onda piloto: simulação, migração e validação funcional com o cliente (clone, merge requests, issues e pipelines).
- Ondas seguintes: uma janela por onda, com a origem em modo somente leitura durante a janela.
- Verificação e contagem origem x destino de cada onda; alertas analisados e registrados; rollback dentro da janela quando necessário.

Fase 3**Otimização**

- Pós-migração: reescrita de referências à origem (merge request por projeto para revisão do cliente), usuários placeholder sem correspondente, runners e integrações no destino e demais itens de tratamento manual da matriz.
- Corte: arquivamento da origem por onda e comunicação aos usuários, com opção de desfazer.

Fase 4**Transferência de conhecimento**

- Revisão dos relatórios das ondas e dos alertas aceitos com a equipe do cliente.
- Entrega do repositório do engajamento com o plano, o registro de decisões e os relatórios.
- Encerramento: remoção dos dados do engajamento no host de migração, remoção do runner, revogação dos tokens, relatório de encerramento e aceite.

9. Entregáveis

Entregável	Descrição	Formato
Relatório de verificação prévia	Origem, destino, método, versão do GitLab Congregate, erros e alertas das verificações por API.	Markdown (artefato do pipeline)
Inventário da origem	Contagens de grupos, projetos, arquivados, vazios e usuários, e planilha por projeto com visibilidade, última atividade, tamanhos de repositório, LFS, pacotes e registry, onda e destino; sem tokens, e-mails ou membros.	Markdown e CSV (artefato do pipeline)
Plano de ondas	Grupos e projetos de cada onda, destino de cada item, grupos intermediários criados e ciclos de execução, validado pelo pipeline.	Repositório do engajamento (YAML e CSV) e Markdown por onda
Relatório de usuários	Quando há criação de usuários: usuário na origem × usuário no destino e situação, sem e-mails. Quando há reatribuição: estado de cada usuário placeholder e aprovadores replicados.	Markdown (artefato do pipeline)
Relatório de simulação de cada onda	Conferência do pacote simulado contra o plano da onda e caminhos de destino livres.	Markdown (artefato do pipeline)
Relatório de cada onda	Verificação no destino, status do Direct Transfer, contagem independente origem × destino item a item, relatório de diferenças do GitLab Congregate e, com cópia de imagens, a lista de imagens copiadas.	Markdown, JSON e HTML (artefato do pipeline)
Relatório de referências à origem	Projetos com ocorrências de URLs, remotos SSH e <code>include</code> apontando para a origem, reescritas e retidas.	Markdown, JSON e CSV (artefato do pipeline)
Relatório de arquivamento da origem	Ação por projeto no corte e estado anterior de cada um; relatório equivalente quando o corte é desfeito.	Markdown (artefato do pipeline)
Relatório de rollback	Quando executado: estado de cada grupo e projeto da onda no destino após o rollback.	Markdown (artefato do pipeline)
Registro de decisões	Decisões do engajamento: ondas, exceções, alertas aceitos, placeholders sem correspondente e momento do corte.	Repositório do engajamento

Entregável	Descrição	Formato
Relatório de encerramento	Resultado das ondas, alertas aceitos, itens de tratamento manual, usuários placeholder pendentes, remoção dos dados do host e do runner e revogação dos tokens.	Documento

10. Premissas

- A instância GitLab self-managed de destino está instalada e em operação, em versão dentro da regra do método escolhido, com as configurações do documento de pré-requisitos.
- O cliente provisiona o host de migração, o runner e a conectividade de rede, e fornece os tokens com os papéis e escopos exigidos.
- Grupos e projetos de cada onda não são alterados na origem depois do início da migração da onda: alterações podem não ser copiadas para o destino (documentação GitLab).
- O nível de assinatura da origem é informado no plano: com o nível errado, o GitLab Congregate ignora sem erro itens pagos como regras de aprovação, webhooks e push rules.
- O rollback só remove o que a onda criou dentro da janela acordada; usuários criados não são removidos.
- Limites padrão do Direct Transfer na instância de destino, configuráveis: 5 GiB por relação baixada, 25 GiB por arquivo descompactado e 6 migrações por minuto por usuário.
- Os relatórios ficam como artefatos dos jobs no projeto do engajamento: 90 dias para a preparação do host e o preflight, 1 ano para os demais.
- Cada engajamento usa uma versão fixa dos componentes do pipeline e do GitLab Congregate; mudança de versão só por merge request.

Fora do escopo

- Destino GitLab.com ou GitLab Dedicated (roadmap).
- Origens Bitbucket Server/Data Center e AWS CodeCommit (roadmap); GitHub, Azure DevOps e Bitbucket Cloud têm ofertas próprias.
- Itens classificados como "não migra" na matriz do documento de pré-requisitos; recriação só por acordo.
- Operação sem acesso à internet no host de migração (air-gapped), que está no roadmap.

Responsabilidades do cliente

- Instância de destino com Direct Transfer habilitado, grupo pai privado ou interno e demais configurações de administrador.

- Host de migração dedicado com Docker, runner shell do engajamento e rede: host → origem, destino e serviços da GitLab; destino → origem por HTTPS.
- Tokens da origem e do destino com os papéis e escopos exigidos, cadastrados como variáveis protegidas do projeto do engajamento.
- Aprovador de cada onda e do rollback, janelas de manutenção e congelamento de escrita na origem durante a janela.
- Decisão sobre usuários: criação, usuários já provisionados, correspondência origem → destino e responsável pelos placeholders sem correspondente.
- Validação funcional dos projetos críticos de cada onda, revisão dos merge requests de reescrita de referências e comunicação do corte aos usuários.

Detalhamento no documento de pré-requisitos PR-MIG-01, enviado após a escolha do serviço.

11. Duração

Prazo definido na proposta, a partir do Levantamento da Plataforma.

A duração depende do volume de dados e do número de projetos, do número de ondas, das janelas de mudança e da prontidão dos pré-requisitos. O Levantamento da Plataforma estima a duração por onda com coeficientes ajustados aos exemplos de duração publicados pela GitLab, que informa não haver fórmula exata; a estimativa é recalibrada com a onda piloto.

Limites de execução do pipeline: o job de migração de cada onda tem limite padrão de 8 horas e cada comando do GitLab Congregate, de 21.600 segundos (6 horas); o pós-importação é aguardado por até 30 minutos por ciclo. Toda estimativa de prazo depende do volume real, dos limites de API e da infraestrutura da origem e do destino, e não é garantia de prazo.

12. Documentos relacionados

- **PR-MIG-01** · Pré-requisitos: Migração GitLab para GitLab self-managed
- **PS-MIG-02** · Migração GitHub para GitLab self-managed
- **PS-MIG-03** · Migração Azure DevOps para GitLab self-managed
- **PS-MIG-04** · Migração Bitbucket Cloud para GitLab self-managed

O3S Consultoria e Tecnologia da Informação Ltda. · contato@pointertech.digital · +55 (61) 99874-4648 ·
pointertech.digital

PS-MIG-01 · versão 1.2 · 1 de outubro de 2026. GITLAB is a trademark of GitLab Inc. in the United States and other countries and regions. A Pointer é parceira GitLab Select.