

DATA SHEET · MIGRAÇÃO

Migração GitHub para GitLab™ self-managed

Repositórios, issues, pull requests e releases do GitHub importados em ondas para a instância GitLab self-managed do cliente, pelo importador GitHub da instância de destino disparado pelo GitLab Congregate

CÓDIGO**PS-MIG-02****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-02

Migração GitHub para GitLab self-managed

Repositórios, issues, pull requests e releases do GitHub importados em ondas para a instância GitLab self-managed do cliente, pelo importador GitHub da instância de destino disparado pelo GitLab Congregate

Migração de repositórios de uma organização do GitHub para a instância GitLab self-managed do cliente, inclusive uma instância implantada pela Pointer. Cada onda usa o **importador GitHub** da instância de destino, recurso nativo da plataforma GitLab, disparado pelo **GitLab Congregate** a partir do pipeline Pointer, seguido das etapas pós-migração do GitLab Congregate, da reatribuição dos usuários placeholder e da verificação da importação por projeto. A origem GitHub.com integra a oferta; GitHub Enterprise Server é atendido sob consulta: escopo e condições definidos em proposta específica, com execução piloto.

1. Para quem

- Organizações no GitHub.com cujos repositórios, issues e pull requests passam para uma instância GitLab self-managed do cliente.
- Organizações com GitHub Enterprise Server (sob consulta: escopo e condições definidos em proposta específica, com execução piloto).
- Equipes com usuários já provisionados no destino por LDAP, SAML ou cadastro, que precisam associar as contribuições do GitHub a esses usuários.

2. Onde executamos

O host de migração fica na rede do cliente (data center próprio ou nuvem), só com conexões de saída. O importador GitHub é executado pela instância GitLab de destino, que busca os dados no GitHub por HTTPS.

3. Escopo do serviço

- Discovery de migração: organização, repositórios, usuários (login no GitHub × usuário no destino), proteções de branch, LFS e workflows do GitHub Actions.
- Plano de ondas por repositório, também por planilha, com destino de cada repositório, 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) do destino.

- Execução das ondas pelo importador GitHub: onda piloto, simulação, importação com aprovação, etapas pós-migração do GitLab Congregate e verificação da importação por projeto.
- Reatribuição em lote dos usuários placeholder, com a planilha de correspondência login do GitHub → usuário no destino.
- Corte por onda: arquivamento dos repositórios no GitHub, com opção de desfazer.
- Rollback da onda dentro da janela acordada.
- Encerramento: remoção dos dados do engajamento no host de migração, remoção do runner, revogação dos tokens e relatório de encerramento.

4. Origens, destino e método suportados

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

- **Sob consulta** Escopo e condições definidos em proposta específica, com execução piloto.

Origem	Destino	Método	Versões	Estado
GitHub.com	GitLab self-managed	Importador GitHub da instância de destino, disparado pelo GitLab Congregate	Serviço GitHub.com; destino GitLab self-managed com a fonte de importação GitHub habilitada	• Disponível
GitHub Enterprise Server	GitLab self-managed	Importador GitHub da instância de destino, pela API, disparado pelo GitLab Congregate	Definidas na proposta e conferidas na execução piloto	◦ Sob consulta

- Destino: somente GitLab self-managed, inclusive instância implantada pela Pointer. GitLab.com e GitLab Dedicated como destino estão no roadmap.
- GitHub Enterprise Server só é importado pela API; a documentação do GitLab Congregate orienta o uso de token gerado por um administrador do site (*site administrator*).
- GitHub Enterprise Server: sob consulta; escopo e condições definidos em proposta específica, com execução piloto.
- Nas versões 19.4 e posteriores, a importação de comentários pelo endpoint único não pode mais ser desligada em importações novas (documentação GitLab).
- Reescrita de referências nos repositórios e contagem independente origem × destino estão disponíveis só com origem GitLab; com origem GitHub, a verificação usa o status e as estatísticas da importação de cada projeto.
- Outras origens: GitLab (PS-MIG-01), 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
Importador GitHub	• Nativo	Recurso nativo da plataforma GitLab, executado pela instância de destino. O GitLab Congregate dispara a importação com a importação de colaboradores ligada e a de anexos desligada.
Ondas de repositórios, também por planilha	• Requer configuração	Cada onda lista repositórios pelo caminho organização/repositório. Destino = grupo pai no destino + organização/repositório, com os grupos criados pelo pipeline; a importação de projetos não migra organizações nem times (documentação GitLab).
Simulação da onda	• Requer configuração	Execução do GitLab Congregate em modo simulação, sem criar nada no destino; caminho de destino já ocupado reprova a onda. Com importadores, a conferência do pacote simulado contra o plano gera alerta.
Etapas pós-migração do GitLab Congregate	• Requer configuração	Depois da importação: aprovações de merge request em nível de projeto, deploy keys, proteção de branch e pipeline de publicação para a branch <code>gh-pages</code> , quando existe.
Reatribuição em lote dos usuários placeholder	• Requer configuração	As contribuições chegam em usuários placeholder, mecanismo nativo da plataforma GitLab. O pipeline reatribui em lote com a planilha login do GitHub → usuário no destino, porque o GitHub só expõe o e-mail público. Sem a opção de reatribuição sem confirmação no destino, cada usuário aceita o pedido.
Verificação da importação por projeto	• Requer configuração	Por API na instância de destino: importação com falha reprova a onda; relações com falha e diferença entre objetos lidos e importados nas estatísticas do importador geram alerta.
Rollback da onda na janela	• Requer configuração	Remove do destino os projetos e grupos da onda criados dentro da janela (padrão 24 horas, de 1 a 720 horas), com exclusão agendada ou permanente; usuários não são removidos.
Arquivamento dos repositórios no GitHub (corte)	• Requer configuração	Arquiva no GitHub os repositórios da onda 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 mascara e-mails. 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 e Redis também fixados, em uma stack de contêineres exclusiva do engajamento no host de migração. Lista a organização, dispara o importador GitHub da instância de destino e executa as etapas pós-migração. Com origem GitHub, roda com 1 processo.

RECURSO NATIVO DA PLATAFORMA GITLAB

Importador GitHub

Importa repositório, issues, pull requests, labels, milestones, releases, wiki, LFS, colaboradores e proteções de branch; executado pela instância de destino.

FERRAMENTA POINTER

psctl

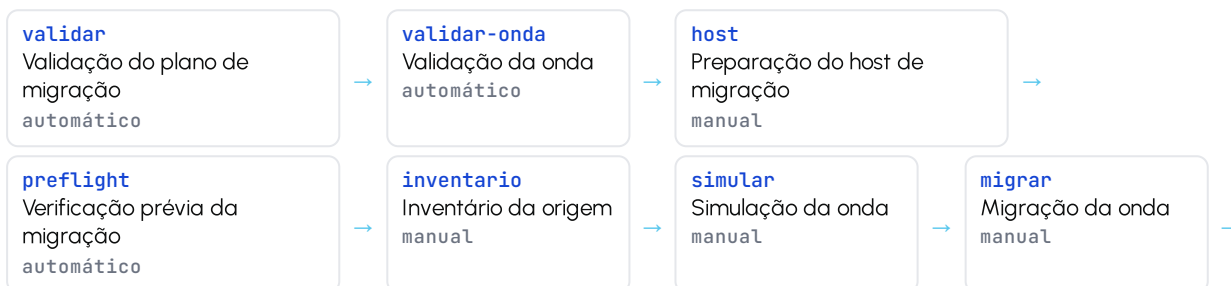
Valida o plano de migração, gera e apaga a configuração do GitLab Congregate em cada job, executa o preflight, resolve cada onda, reatribui os usuários placeholder, arquiva a origem, verifica cada importação e gera os relatórios, com saídas sanitizadas.

COMPONENTE OFICIAL DA GITLAB

GitLab Runner

Runner de executor shell, exclusivo do projeto do engajamento, instalado no host de migração do cliente.

7. Como o pipeline Pointer executa



remover-host

Encerramento: remoção dos dados do host
manual

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**).

1. **Validação do plano de migração** (automática a cada alteração): origem, destino, host, usuários, ondas e variáveis exigidas.
2. **Preparação do host de migração**: stack do GitLab Congregate com versões fixas, sem portas publicadas.
3. **Verificação prévia da migração** (automática): token e versão do destino, fonte de importação GitHub habilitada, grupo pai no destino e configuração do GitLab Congregate.
4. **Inventário da origem**: repositórios e usuários do escopo e repositórios fora de ondas.
5. Por onda: **Simulação da onda** → **Migração da onda** (com aprovação), com etapas pós-migração, reatribuição dos usuários placeholder e verificação da importação por projeto.
6. Depois da onda: **Corte: origem somente leitura** (e **Desfazer o corte**) e **Reversão da onda (rollback)** dentro da janela.
7. **Encerramento: remoção dos dados do host**.

8. Metodologia

Assessment → Implantação → Otimização → Transferência de conhecimento.

Fase 1**Assessment**

- Kick-off: escopo (organização, repositórios, volumes), janelas, congelamento de escrita no GitHub por onda, aprovadores de cada onda e do rollback.
- Discovery de migração: login de cada pessoa no GitHub x usuário no destino, proteções de branch, LFS, workflows do GitHub Actions e integrações; respostas no registro de decisões do engajamento.
- Envio do documento de pré-requisitos (PR-MIG-02) 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, sem repositório fora de onda sem decisão registrada.

Fase 2**Implantação**

- Onda piloto: simulação, importação e validação funcional com o cliente (clone, merge requests, issues).
- Ondas seguintes: uma janela por onda, com congelamento de escrita no GitHub durante a janela.
- Verificação da importação de cada repositório e reatribuição dos usuários placeholder; alertas analisados e registrados; rollback dentro da janela quando necessário.

Fase 3**Otimização**

- Pós-migração: usuários placeholder sem correspondente, *status checks* obrigatórios recriados como verificações externas, workflows do GitHub Actions convertidos para GitLab CI/CD, runners e integrações no destino, conforme os responsáveis do plano.
- Corte: arquivamento dos repositórios no GitHub 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	Destino, fonte de importação, versão do GitLab Congregate, erros e alertas.	Markdown (artefato do pipeline)
Inventário da origem	Repositórios e usuários do escopo, onda e destino de cada repositório e repositórios fora de ondas; sem tokens, e-mails ou membros.	Markdown e CSV (artefato do pipeline)
Plano de ondas	Repositórios de cada onda e destino de cada um, validado pelo pipeline.	Repositório do engajamento (YAML e CSV) e Markdown por onda
Relatório de redistribuição de usuários	Estado de cada usuário placeholder da onda (redistribuído, aguardando aceite ou sem correspondente), sem e-mails.	Markdown (artefato do pipeline)
Relatório de simulação de cada onda	Resultado da simulação e caminhos de destino livres.	Markdown (artefato do pipeline)
Relatório de cada onda	Existência de cada repositório no destino, status da importação e estatísticas do importador por projeto (objetos lidos x importados) e relações com falha.	Markdown e JSON (artefato do pipeline)
Relatório de arquivamento da origem	Ação por repositório no corte e estado anterior; relatório equivalente quando o corte é desfeito.	Markdown (artefato do pipeline)
Relatório de rollback	Quando executado: estado de cada projeto e grupo 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
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, com a fonte de importação GitHub habilitada e as demais configurações do documento de pré-requisitos.
- Os usuários do destino já existem (LDAP, SAML ou cadastro): o pipeline não cria usuários a partir do GitHub.
- A organização no GitHub não restringe a instância de destino por política de acesso de aplicações de terceiros (documentação GitLab).
- Repositórios da onda não recebem alterações no GitHub durante a janela da onda.
- Anexos em Markdown não são importados: os links continuam apontando para o GitHub e deixam de funcionar se os anexos forem removidos do GitHub.
- O rollback só remove o que a onda criou dentro da janela acordada; usuários não são removidos.
- 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).
- Organizações e times do GitHub como estrutura: a importação de projetos não migra organizações nem grupos (documentação GitLab).
- Itens classificados como "não migra" na matriz do documento de pré-requisitos; recriação só por acordo.
- Origens Bitbucket Server/Data Center e AWS CodeCommit (roadmap).
- Operação sem acesso à internet no host de migração (air-gapped), que está no roadmap.

Responsabilidades do cliente

- Instância de destino com a fonte de importação GitHub habilitada, grupo pai privado ou interno e decisão do administrador sobre a reatribuição sem confirmação.
- Host de migração dedicado com Docker, runner shell do engajamento e rede: host → GitHub, destino e serviços da GitLab; destino → GitHub por HTTPS.
- Token clássico do GitHub e token do destino com os escopos e papéis exigidos, cadastrados como variáveis protegidas do projeto do engajamento.
- Usuários existentes no destino e planilha de correspondência login do GitHub → usuário no destino.

- Aprovador de cada onda e do rollback, janelas e congelamento de escrita no GitHub durante a janela.
- Validação funcional dos repositórios críticos de cada onda e comunicação do corte aos usuários.

Detalhamento no documento de pré-requisitos PR-MIG-02, enviado após a escolha do serviço.

11. Duração

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

A duração depende do número de repositórios e de ondas, das janelas de mudança e da prontidão dos pré-requisitos.

Estimativa oficial citada como referência: a documentação GitLab informa que a importação do repositório kubernetes/kubernetes, com cerca de 80 mil pull requests, 45 mil issues e 1,5 milhão de comentários, levou 76 horas. A API do GitHub tem limite, em geral, de 5.000 requisições por hora, e com origem GitHub o GitLab Congregate roda com 1 processo (no GitLab Congregate 8.6.0, a listagem do GitHub com 2 ou mais processos falha). A duração real depende do volume de cada repositório, dos limites de API e da infraestrutura do destino; o exemplo não é garantia de prazo.

12. Documentos relacionados

- **PR-MIG-02** · Pré-requisitos: Migração GitHub para GitLab self-managed
- **PS-MIG-01** · Migração GitLab 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-02 · 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.