

PRÉ-REQUISITOS · MIGRAÇÃO

Pré-requisitos: Migração GitHub para GitLab™ self-managed

Documento enviado após a escolha do serviço PS-MIG-02: o que o cliente provisiona, as credenciais que entrega à Pointer, as informações a enviar e o checklist de prontidão.

CÓDIGO

PR-MIG-02

VERSÃO

1.2

TIPO

Pré-requisitos

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.



GitLab
Select
Partner

PRÉ-REQUISITOS · PR-MIG-02 · OFERTA PS-MIG-02

Pré-requisitos: Migração GitHub para GitLab self-managed

Documento enviado ao cliente depois da escolha do serviço **PS-MIG-02 — Migração GitHub para GitLab self-managed**. Lista o que o cliente provisiona, as credenciais que entrega, as informações a enviar à Pointer, o checklist de prontidão e a matriz do que é migrado. O prazo de devolução das informações e do checklist é definido no kick-off.

Os itens do checklist marcados como **Preflight automático** são conferidos pelo pipeline Pointer contra a instância de destino real antes da primeira onda; os demais são conferidos na reunião de prontidão ou pelo próprio cliente. Nenhuma onda é executada com erro no preflight.

Conteúdo

Origem e destino suportados

Configuração da instância de destino

Organização e repositórios no GitHub

Host de migração

Runner do engajamento

Rede

Usuários e contribuições

Governança da migração

Critérios de aceite por onda

Entrega das credenciais

Credenciais que o cliente entrega à Pointer

Informações a enviar

Matriz de itens migrados

Checklist de prontidão

1. Origem e destino suportados

Item	Suportado
Origem	GitHub.com (organização). GitHub Enterprise Server: sob consulta (escopo e condições definidos em proposta específica, com execução piloto), só pela API
Destino	Somente GitLab self-managed do cliente, inclusive instância implantada pela Pointer
Método	Importador GitHub da instância de destino, disparado pelo GitLab Congregate, seguido das etapas pós-migração do GitLab Congregate. A instância de destino busca os dados no GitHub
Versões	GitHub.com: serviço. GitHub Enterprise Server: versões definidas na proposta e conferidas na execução piloto. Destino: nas versões 19.4 e posteriores, a importação de comentários pelo endpoint único não pode ser desligada em importações novas
Limite de API do GitHub	Em geral 5.000 requisições por hora (documentação GitLab); 15.000 com OAuth app do GitHub Enterprise Cloud, fluxo não usado pelo pipeline
Destinos fora do escopo	GitLab.com e GitLab Dedicated como destino: roadmap
Outras origens	GitLab (PS-MIG-01), Azure DevOps (PS-MIG-03), Bitbucket Cloud (PS-MIG-04); Bitbucket Server/Data Center e AWS CodeCommit: roadmap

- Com origem GitHub, o pipeline verifica cada importação pelo status e pelas estatísticas do importador; a contagem independente origem x destino e a reescrita de referências nos repositórios estão disponíveis só com origem GitLab.

2. Configuração da instância de destino

Item	Especificação	Obrigatório
Fonte de importação GitHub	Habilitada nas configurações de importação e exportação da instância (<i>Admin</i> → <i>Settings</i> → <i>General</i> → <i>Import and export settings</i> , fontes de importação). Em GitLab self-managed nenhuma fonte de importação vem habilitada por padrão. O preflight reprova se ausente	Sim
Grupo pai	Grupo existente, privado ou interno, que recebe a migração. Destino de cada repositório = grupo pai + organização/repositório. O preflight reprova grupo inexistente ou público	Sim
Reatribuição sem confirmação de cada usuário	<i>Skip confirmation when administrators reassign placeholder users</i> . decisão do administrador da instância. Ligada, a reatribuição em lote é imediata; desligada, cada usuário aceita o pedido antes de as contribuições aparecerem. O preflight alerta se estiver desligada com a reatribuição no plano	Condicional: reatribuição em lote no plano

Item	Especificação	Obrigatório
Assentos	Colaboradores do GitHub importados como membros podem consumir assentos da assinatura do destino (documentação GitLab)	Condicional: colaboradores no GitHub

3. Organização e repositórios no GitHub

- A organização não pode ter política de acesso de aplicações de terceiros (*third-party application access policy*) que restrinja a instância de destino (documentação GitLab).
- *Custom repository roles* do GitHub Enterprise Cloud não são suportados e causam importação parcial (documentação GitLab).
- Colaboradores diretos são importados com o mapeamento de papéis Read → Guest, Triage → Reporter, Write → Developer, Maintain → Maintainer e Admin → Owner; *outside collaborators* não são importados.
- O GitHub só expõe o e-mail público dos usuários: a correspondência com os usuários do destino é feita pela planilha login do GitHub → usuário no destino.
- A regra *Require status checks to pass before merging* não é importada; é recriada como verificação externa (*external status checks*) no destino.
- Durante a janela da onda, os repositórios da onda não recebem alterações no GitHub.

4. Host de migração

Item	Especificação	Obrigatório
Máquina	Linux x86_64 dedicada ao engajamento, sem outros serviços: o usuário do runner participa do grupo docker , o que equivale a root no host. Um host por cliente	Sim
Recursos de partida	4 vCPU, 16 GB de RAM e 100 GB de disco SSD (imagem do GitLab Congregate com cerca de 3,3 GB; o banco MongoDB da ferramenta cresce com a listagem da origem). A documentação do GitLab Congregate (página <i>Migration Workflows</i>) pede, no mínimo, ambiente com Docker e 4 CPU, 8 GB de RAM e 20 GB de disco	Sim
Contêineres	Docker Engine e Docker Compose v2; a ausência reprova o início de cada job	Sim
Swap	2 GB de swap, conforme a documentação do GitLab Congregate	Não
Certificados	Certificado válido no destino. CA corporativa em arquivo no host (arquivo inexistente reprova o job), somada às raízes do contêiner do GitLab Congregate	Condicional: CA interna

Item	Especificação	Obrigatório
Proxy	Proxy de saída informado no plano de migração e configurado também no daemon Docker	Condicional: proxy corporativo
Acompanhamento	A interface do GitLab Congregate e o Flower ficam só em <code>127.0.0.1</code> do host; o acompanhamento é por túnel SSH (usuário@host)	Não

- No encerramento, o pipeline remove a stack e todos os dados do engajamento no host (listagens, logs e resultados).

5. Runner do engajamento

Item	Especificação	Obrigatório
GitLab Runner	Instalado no host de migração, executor shell, registrado no projeto do engajamento na GitLab.com da Pointer com o token de registro criado nesse projeto	Sim
Configuração	Runner de projeto com a tag do engajamento; <i>Run untagged jobs</i> desmarcado; <i>Protected</i> e <i>Lock to current projects</i> marcados; tempo máximo de job de 8 horas ou mais	Sim
Usuário do runner	<code>gitlab-runner</code> no grupo <code>docker</code> ; em Ubuntu, remover <code>/home/gitlab-runner/.bash_logout</code> (o <code>clear_console</code> encerra jobs shell)	Sim
Encerramento	Runner removido do projeto e desinstalado do host	Sim

6. Rede

Conexão	Protocolo e observação	Obrigatório
Host de migração → GitHub (API e repositórios)	HTTPS; GitHub.com ou o endereço do GitHub Enterprise Server	Sim
Host de migração → instância de destino	HTTPS	Sim
Host de migração → gitlab.com	HTTPS: runner e projeto do engajamento	Sim
Host de migração → registry.gitlab.com	HTTPS: imagens do GitLab Congregate e da plataforma Pointer. Para pull de registry.gitlab.com, a documentação da GitLab.com lista também <code>*.storage.googleapis.com</code> e <code>*.cdn.registry.gitlab-static.net</code>	Sim

Conexão	Protocolo e observação	Obrigatório
Host de migração → Docker Hub	HTTPS: imagens do MongoDB e do Redis	Sim
Instância de destino → GitHub	HTTPS: o importador GitHub é executado pela instância de destino	Sim
Estação do engenheiro → host de migração	SSH, para o túnel de acompanhamento	Não

- O host de migração só faz conexões de saída. Espelhos internos equivalentes podem substituir gitlab.com, registry.gitlab.com e Docker Hub.
- Operação sem acesso à internet (air-gapped) está no roadmap.

7. Usuários e contribuições

- Contribuições chegam em usuários placeholder na instância de destino (mapeamento posterior à migração, padrão desde a versão 17.8).
- O pipeline não cria usuários a partir do GitHub: os usuários do destino já existem, provisionados por LDAP, SAML ou cadastro.
- Reatribuição em lote pelo pipeline, com a planilha login do GitHub → usuário no destino, que vale antes da correspondência por e-mail (só o e-mail público do GitHub) ou por username.
- Reatribuição pedida pelo Owner do grupo de topo; com a opção de reatribuição sem confirmação ligada no destino, é imediata.
- A conta dona do token do destino é a conta da importação.

8. Governança da migração

- Aprovador de cada onda e do rollback, do lado do cliente.
- Janela de manutenção de cada onda e congelamento de escrita no GitHub durante a janela.
- Responsável pelos usuários placeholder sem correspondente (Owner do grupo de topo) e prazo.
- Momento do corte (arquivamento dos repositórios no GitHub) e comunicação aos usuários.
- Responsável por cada item de tratamento manual da matriz, inclusive a conversão dos workflows do GitHub Actions.

9. Critérios de aceite por onda

- Relatório da onda sem erro: todo repositório existe no destino e nenhuma importação terminou com falha.

- Alertas (relações com falha, diferença entre objetos lidos e importados nas estatísticas do importador) analisados e aceitos ou corrigidos, com registro no registro de decisões.
- Com reatribuição: todos os usuários reatribuídos ou aguardando aceite, se o destino exigir confirmação.
- Validação funcional pelo cliente nos repositórios críticos da onda: clone, merge requests e issues.

10. Entrega das credenciais

Cada credencial é cadastrada como variável de CI/CD do projeto do engajamento, com as opções *Protected* e *Masked and hidden* e escopo restrito ao ambiente de migração, que é protegido e exige aprovação para a migração e o rollback de cada onda. O pipeline lista, sem valores, as variáveis exigidas pelo plano, e o preflight reprova se faltar alguma. A configuração do GitLab Congregate com os tokens só existe durante cada job no host de migração.

Tokens nunca são colados em chat, issue ou arquivo; se isso ocorrer, o token é revogado e outro é emitido. No encerramento, os tokens usados na origem e no destino são revogados e as contas de administrador criadas para o engajamento são desativadas.

11. Credenciais que o cliente entrega à Pointer

Nenhum valor secreto é enviado por e-mail, chat, issue ou arquivo. Cada credencial é cadastrada como variável de CI/CD protegida no projeto do engajamento, com escopo do ambiente, e revogada no encerramento.

Credencial	Finalidade	Permissões	Formato e validade	Obrigatório
Token de acesso pessoal da instância de destino	Usado pelo GitLab Congregate (disparo do importador e etapas pós-migração), pela verificação das importações, pela reatribuição de usuários e pelo rollback.	Escopo <code>api</code> , e também <code>admin_mode</code> quando o Admin Mode estiver habilitado na instância. Usuário administrador da instância: sem administrador o preflight alerta e não confere as configurações de importação. A importação exige, no mínimo, papel Maintainer ou Owner no grupo de destino (documentação GitLab).	Personal access token da instância GitLab de destino, cadastrado como variável de CI/CD <code>PS_MIG_DESTINO_TOKEN</code> (Protected, Masked and hidden, escopo do ambiente de migração) Expiração na data estimada da última onda, conforme a documentação do GitLab Congregate; o preflight alerta com menos de 7 dias. Revogado no encerramento.	Sim

Credencial	Finalidade	Permissões	Formato e validade	Obrigatório
Token de acesso pessoal clássico do GitHub	Usado pelo GitLab Congregate (listagem e disparo do importador na instância de destino), pela reatribuição de usuários e pelo arquivamento e desarquivamento dos repositórios no corte.	Escopos <code>repo</code> e <code>read:org</code> (<code>read:org</code> também é exigido para importar colaboradores e objetos LFS). Papel Write ou Maintain nos repositórios migrados para importar os colaboradores; sem ele, a importação de colaboradores é pulada (documentação GitLab). O runbook do GitLab Congregate descreve token da organização com privilégios de owner. GitHub Enterprise Server: token gerado por administrador do site. O corte arquiva os repositórios com este token.	Personal access token clássico do GitHub (o importador aceita só tokens clássicos), cadastrado como variável de CI/CD <code>PS_MIG_ORIGEM_TOKEN</code> (Protected, Masked and hidden, escopo do ambiente de migração) Expiração na data estimada da última onda. Revogado no encerramento.	Sim

12. Informações a enviar

Informações sem sigilo, devolvidas pelo cliente antes da reunião de prontidão. Os campos marcados como obrigatórios são conferidos pela validação automática do pipeline.

Origem

Informação	Exemplo	Obrigatório	Observação
Tipo de origem	GitHub.com	Sim	GitHub.com ou GitHub Enterprise Server (sob consulta).
URL da API do GitHub	https://api.github.com	Sim	No GitHub Enterprise Server, a URL da API do servidor.
Organização	orgao-exemplo	Condicional: GitHub.com	Define o escopo da listagem.

Destino

Informação	Exemplo	Obrigatório	Observação
URL da instância de destino	https://gitlab.exemplo.gov.br	Sim	
Versão e nível de assinatura do destino	19.4.1, Premium	Sim	
Grupo pai no destino	migrados-github	Sim	Grupo privado ou interno existente.

Informação	Exemplo	Obrigatório	Observação
Reatribuição sem confirmação habilitada	Sim	Condicional: reatribuição em lote	Decisão do administrador do destino.

Host de migração e rede

Informação	Exemplo	Obrigatório	Observação
Nome e sistema operacional do host de migração	host-migracao.exemplo.gov.br, Ubuntu 24.04	Sim	
Proxy de saída	https: http://proxy.exemplo.gov.br:3128; exceções: .exemplo.gov.br	Condicional: proxy corporativo	HTTPS, HTTP e exceções.
Caminho da CA corporativa no host	/etc/pki/ca-trust/source/anchors/ac-exemplo.pem	Condicional: CA interna	Caminho absoluto de arquivo existente no host.
Acesso SSH para acompanhamento	usuario@host-migracao.exemplo.gov.br	Não	Túnel para a interface do GitLab Congregate.
Confirmação de conectividade destino → GitHub por HTTPS	Sim	Sim	

Usuários

Informação	Exemplo	Obrigatório	Observação
Planilha de correspondência de usuários	usuarios-mapa.csv (colunas origem, destino)	Condicional: reatribuição em lote	Login no GitHub → username no destino.
Campo de correspondência para quem não está na planilha	Username	Não	E-mail (só o e-mail público do GitHub) ou username.
Origem dos usuários no destino	LDAP	Sim	LDAP, SAML ou cadastro.
Responsável pelos placeholders sem correspondente e prazo	Owner do grupo de topo; 10 dias	Sim	

Ondas

Informação	Exemplo	Obrigatório	Observação
Repositórios de cada onda	piloto: orgao-exemplo/app-web, orgao-exemplo/lib-util	Sim	Caminho organização/repositório; o mesmo repositório não se repete entre ondas. Também por planilha (colunas onda e projeto).

Informação	Exemplo	Obrigatório	Observação
Nome, data e descrição de cada onda	piloto; 10/11/2026; repositórios de baixo risco	Sim	A primeira onda é a piloto.
Janela de manutenção de cada onda	10/11/2026, 19h às 23h	Sim	
Aprovador de cada onda e do rollback	Nome e cargo	Sim	
Janela de rollback	24 horas	Não	De 1 a 720 horas; padrão 24.
Confirmação do congelamento de escrita no GitHub durante a janela	Sim	Sim	

Pós-migração e corte

Informação	Exemplo	Obrigatório	Observação
Arquivamento dos repositórios no GitHub no corte	Sim, 5 dias após cada onda	Sim	Sim ou não, e momento.
Responsáveis pelos itens de tratamento manual	Conversão do GitHub Actions: equipe de plataforma	Sim	Conforme a matriz.

Contatos

Informação	Exemplo	Obrigatório	Observação
Patrocinador	Nome, e-mail, telefone	Sim	
Ponto focal técnico	Nome, e-mail, telefone	Sim	
Owner da organização no GitHub	Nome, e-mail	Sim	
Administrador da instância de destino	Nome, e-mail	Sim	
Infraestrutura, rede e segurança	Nome, e-mail	Sim	Host, firewall, proxy e certificados.

13. Matriz de itens migrados

- Nativo** Migrado pelo método oficial (importador GitHub da instância de destino), disparado pelo pipeline
- Pipeline Pointer** Tratado pelo pipeline Pointer após a importação (etapas pós-migração do GitLab Congregate)
- Ajuste manual** Exige ajuste manual após a migração (com responsável definido no plano)
- Não migra** Não é migrado; recriação fora do escopo, salvo acordo

O tratamento de cada item segue a documentação oficial do método de migração e é conferido na onda piloto do engajamento.

Repositório

Item	Tratamento	Observação
Repositório Git (todas as branches e tags) e descrição	Nativo	
Branches de forks ligadas a pull requests abertos	Nativo	Importadas com o padrão <code>GH-SHA-username/pull-request-number/fork-name/branch</code> .
Objetos Git LFS	Nativo	Exige token com <code>read:org</code> quando há LFS.
Wiki	Nativo	
Releases (texto das notas)	Nativo	
Regras de proteção de branch	Nativo	Mapeadas para proteção de branch e configurações: conversa resolvida, pull request obrigatório, commits assinados (push rule), force push e Code Owners. O GitLab Congregate também aplica a proteção de branch no pós-migração. A matriz do GitLab Congregate lista proteções como não suportadas pelo importador; adotada a documentação GitLab.
Regra <i>Require status checks to pass before merging</i>	Ajuste manual	Não é importada; recriada como verificação externa (<i>external status checks</i>) no destino (documentação GitLab).
GitHub Pages	Pipeline Pointer	O GitLab Congregate grava na branch <code>gh-pages</code> , quando existe, um <code>.gitlab-ci.yml</code> de publicação de HTML simples. A matriz do GitLab Congregate lista Pages como ainda não suportado; adotado o comportamento do código do GitLab Congregate 8.6.0.

Pull requests → merge requests

Item	Tratamento	Observação
Pull requests (comentários, reviews, comentários de review, respostas, sugestões, revisores atribuídos, quem fez o merge)	Nativo	Comentários de pull request anteriores a 2017 viram threads separadas.
Aprovações de merge request em nível de projeto	Pipeline Pointer	Etapa fixa do pós-migração do GitLab Congregate.

Issues e planejamento

Item	Tratamento	Observação
Issues e comentários	Nativo	
Labels e milestones	Nativo	

CI/CD e automação

Item	Tratamento	Observação
Deploy keys	Pipeline Pointer	O importador não importa; recriadas pelo pós-migração do GitLab Congregate.
Workflows do GitHub Actions	Ajuste manual	Conversão para GitLab CI/CD, fora da importação.

Usuários, organização e acesso

Item	Tratamento	Observação
Colaboradores diretos do repositório	Nativo	Importados com a importação de colaboradores ligada pelo GitLab Congregate; exigem papel Write ou Maintain do token no GitHub e podem consumir assentos. <i>Outside collaborators</i> não são importados.
Organização e times do GitHub	Não migra	A importação de projetos não migra organizações nem grupos; os grupos de destino são criados pelo pipeline (grupo pai + caminho na origem).
Contas de usuário, chaves SSH e GPG e tokens	Não migra	O pipeline não cria usuários de origem GitHub; os usuários do destino vêm de LDAP, SAML ou cadastro.

Conteúdo não migrado

Item	Tratamento	Observação
Anexos em Markdown (comentários, issues, pull requests, releases)	Não migra	O GitLab Congregate 8.6.0 dispara a importação com anexos desligados; os links continuam apontando para o GitHub e deixam de funcionar quando os anexos são removidos do GitHub.
Demais recursos do GitHub fora da lista de dados importados	Não migra	A documentação GitLab lista os itens importados; a documentação do GitLab Congregate presume não suportado o que não consta da matriz dele.

Notas da matriz

- **A confirmar na onda piloto:** eventos de issues e de pull requests (a documentação GitLab informa que podem ser importados como item adicional; a API de importação lista como etapas opcionais só anexos, colaboradores e comentários pelo endpoint único); webhooks (não importados pelo importador; o GitLab Congregate informa que ainda não suporta).
- Referências: a instância de destino cria links só para issues citadas em comentários (o GitHub usa # para issues e pull requests) e não cria links em descrições.
- GitHub Enterprise Server: toda a matriz é conferida na onda piloto.

14. Checklist de prontidão

Itens conferidos antes da execução. "Preflight automático" indica verificação feita pelo pipeline contra o ambiente real; os demais são conferidos na reunião de prontidão ou pelo cliente.

☐ Item	Verificado por
☐ Docker Engine e Docker Compose v2 presentes no host; CA informada existente no host (verificado na preparação do host).	Preflight automático
☐ Variáveis de CI/CD exigidas pelo plano cadastradas.	Preflight automático
☐ Instância de destino acessível e token do destino válido, com papel de administrador.	Preflight automático
☐ Token do destino com validade maior que 7 dias.	Preflight automático
☐ Fonte de importação GitHub habilitada no destino.	Preflight automático
☐ Grupo pai no destino existente e privado ou interno.	Preflight automático
☐ Reatribuição sem confirmação habilitada no destino, quando a reatribuição em lote está no plano (alerta).	Preflight automático
☐ Configuração do GitLab Congregate validada.	Preflight automático
☐ Instância de destino alcança o GitHub por HTTPS.	Reunião de prontidão
☐ Token clássico do GitHub com <code>repo</code> e <code>read:org</code> e papel suficiente para importar colaboradores e arquivar os repositórios.	Reunião de prontidão
☐ Organização sem política de acesso de aplicações de terceiros que restrinja o destino.	Reunião de prontidão
☐ Runner online, com a tag do engajamento, protegido, travado no projeto e com tempo máximo de job de 8 horas ou mais.	Reunião de prontidão
☐ Plano de ondas com onda piloto, janelas, aprovadores e janela de rollback definidos.	Reunião de prontidão
☐ Usuários existentes no destino e planilha de correspondência login do GitHub → usuário no destino conferida.	Reunião de prontidão
☐ GitHub Enterprise Server: onda piloto definida.	Reunião de prontidão
☐ Host de migração provisionado conforme a especificação, com saída HTTPS para GitHub, destino, gitlab.com, registry.gitlab.com e Docker Hub.	Cliente
☐ Tokens do GitHub e do destino emitidos com escopos, papéis e validade exigidos e cadastrados como variáveis protegidas.	Cliente
☐ Congelamento de escrita comunicado aos usuários do GitHub para a janela de cada onda.	Cliente

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

PR-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.