

DATA SHEET · MIGRAÇÃO

Migração Azure DevOps para GitLab™ self-managed

Repositórios Git, pull requests e work items de backlog do Azure DevOps Services migrados em ondas para a instância GitLab self-managed do cliente, por exportação montada pelo GitLab Congregate e importação por arquivo

CÓDIGO**PS-MIG-03****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-03

Migração Azure DevOps para GitLab self-managed

Repositórios Git, pull requests e work items de backlog do Azure DevOps Services migrados em ondas para a instância GitLab self-managed do cliente, por exportação montada pelo GitLab Congregate e importação por arquivo

Migração de repositórios Git, pull requests e work items de backlog de projetos do Azure DevOps Services para a instância GitLab self-managed do cliente, inclusive uma instância implantada pela Pointer. A plataforma GitLab não tem importador nativo de Azure DevOps: o **GitLab Congregate** lê o Azure DevOps no host de migração, monta arquivos de exportação no formato GitLab e a instância de destino os importa por arquivo, recurso nativo da plataforma GitLab, em ondas disparadas pelo pipeline Pointer. A autoria é associada pelo e-mail no momento da importação, com os usuários criados no destino antes das ondas. Azure DevOps Server é atendido sob consulta: escopo e condições definidos em proposta específica, com execução piloto.

1. Para quem

- Organizações no Azure DevOps Services cujos repositórios Git, pull requests e work items de backlog passam para uma instância GitLab self-managed do cliente.
- Organizações com Azure DevOps Server (sob consulta: escopo e condições definidos em proposta específica, com execução piloto).
- Equipes que precisam manter a autoria de pull requests, comentários e work items associada aos usuários do destino.

2. Onde executamos

O host de migração fica na rede do cliente (data center próprio ou nuvem), só com conexões de saída: lê o Azure DevOps e envia os arquivos de exportação à instância GitLab de destino, que importa por arquivo.

3. Escopo do serviço

- Discovery de migração: organização, projetos e repositórios, usuários e e-mails, tipos de work item, políticas de branch, pipelines YAML, variable groups, wikis e integrações.
- Plano de ondas por projeto do Azure DevOps, 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.

- Criação dos usuários no destino antes das ondas pelo GitLab Congregate, com conferência de cada um pelo e-mail, ou uso de usuários já provisionados por LDAP ou SAML com o mesmo e-mail.
- Execução das ondas: onda piloto, simulação, exportação e importação por arquivo com aprovação, pós-importação dos anexos e verificação da importação por projeto.
- 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
Azure DevOps Services	GitLab self-managed	Exportação montada pelo GitLab Congregate e importação por arquivo na instância de destino	Serviço Azure DevOps Services; destino GitLab self-managed com a fonte de importação <i>GitLab export</i> habilitada	• Disponível
Azure DevOps Server	GitLab self-managed	Exportação montada pelo GitLab Congregate e importação por arquivo na instância de destino	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.
- Repositórios TFVC estão fora do escopo: a documentação GitLab informa que não há ferramenta da GitLab para migrar de TFVC para Git.
- Azure DevOps Server: sob consulta; escopo e condições definidos em proposta específica, com execução piloto. O runbook do GitLab Congregate prevê uma aplicação própria para listar os usuários.
- O work item Epic do Azure DevOps não vira issue nem épico nesse fluxo.
- Com origem Azure DevOps não há arquivamento da origem, reescrita de referências nos repositórios nem contagem independente origem × destino; a verificação usa o status e as estatísticas da importação de cada projeto.
- Outras origens: GitLab (PS-MIG-01), GitHub (PS-MIG-02) 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
Importação por arquivo na instância de destino	• Nativo	Recurso nativo da plataforma GitLab (fonte de importação <i>GitLab export</i>), executado pela instância de destino a partir dos arquivos montados pelo GitLab Congregate.
Leitura do Azure DevOps e montagem da exportação	• Requer configuração	O GitLab Congregate lê repositórios, pull requests e work items pelas APIs do Azure DevOps e por Git, no host de migração, e monta os arquivos de exportação. Work items de backlog viram issues com título, descrição convertida de HTML para Markdown, estado (New e Active → aberta; Resolved e Closed → fechada), autor, responsável, comentários, e tipo e tags como labels.
Ondas por projeto do Azure DevOps, também por planilha	• Requer configuração	Cada onda lista projetos do Azure DevOps. Destino = grupo pai no destino + projeto do Azure DevOps + repositório, com o grupo do projeto criado pelo pipeline antes da importação.
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. A conferência do pacote simulado contra o plano gera alerta.
Criação dos usuários antes das ondas	• Requer configuração	Não há usuários placeholder: a autoria e as memberships são associadas pelo e-mail no momento da importação. O GitLab Congregate cria no destino os usuários dos times do projeto no Azure DevOps e o pipeline confere cada um pelo e-mail antes da primeira onda.
Pós-importação dos anexos	• Requer configuração	Etapa fixa do GitLab Congregate: baixa do Azure DevOps os anexos de pull requests e work items, envia à instância de destino e reescreve os links.
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 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.

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 do Azure DevOps, cria os usuários, monta a exportação, dispara a importação por arquivo e executa o pós-importação dos anexos.

RECURSO NATIVO DA PLATAFORMA GITLAB

Importação de projetos por arquivo

Importa no destino os arquivos de exportação montados pelo GitLab Congregate.

FERRAMENTA POINTER

psctl

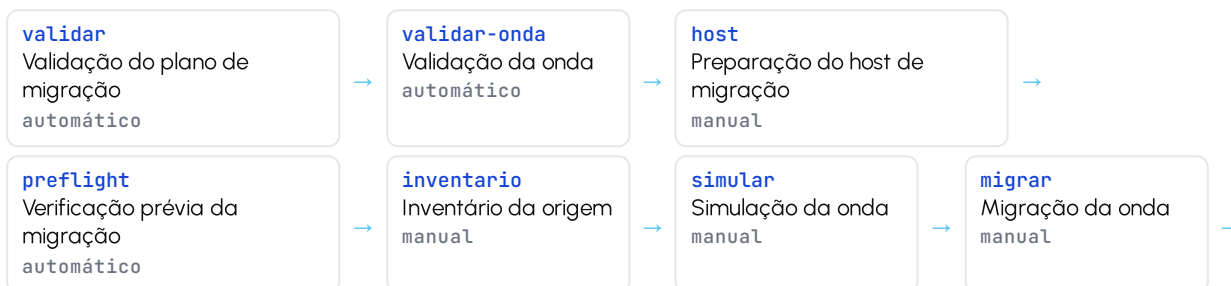
Valida o plano de migração, gera e apaga a configuração do GitLab Congregate em cada job, executa o preflight, cria os grupos dos projetos do Azure DevOps, confere os usuários, verifica cada importação e gera os relatórios, com saídas sanitizadas.

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 *GitLab export* habilitada, grupo pai no destino e configuração do GitLab Congregate.
4. **Inventário da origem**: projetos, repositórios e usuários do escopo e projetos fora de ondas.
5. **Criação dos usuários antes das ondas**: usuários do Azure DevOps criados no destino e conferidos pelo e-mail.
6. Por onda: **Simulação da onda** → **Migração da onda** (com aprovação), com pós-importação dos anexos e verificação da importação por projeto.
7. Depois da onda, quando necessário: **Reversão da onda (rollback)** dentro da janela.
8. **Encerramento: remoção dos dados do host**.

8. Metodologia

Assessment → Implantação → Otimização → Transferência de conhecimento.

Fase 1**Assessment**

- Kick-off: escopo (organização, projetos, repositórios, volumes), janelas, congelamento de escrita no Azure DevOps por onda, aprovadores de cada onda e do rollback.
- Discovery de migração: usuários e e-mails no Azure DevOps x no destino, tipos de work item em uso, políticas de branch, pipelines YAML, variable groups, wikis, feeds do Azure Artifacts e imagens do Azure Container Registry; respostas no registro de decisões do engajamento.
- Envio do documento de pré-requisitos (PR-MIG-03) 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 projeto fora de onda sem decisão registrada.

Fase 2**Implantação**

- Criação dos usuários no destino antes da primeira onda, com todos os usuários conferidos pelo e-mail, ou provisionamento por LDAP ou SAML com o mesmo e-mail.
- Onda piloto: simulação, importação e validação funcional com o cliente (clone, merge requests, issues e autoria).
- Ondas seguintes: uma janela por onda, com congelamento de escrita no Azure DevOps durante a janela.
- Verificação da importação de cada projeto; alertas analisados e registrados; rollback dentro da janela quando necessário.

Fase 3**Otimização**

- Pós-migração: políticas de branch, pipelines YAML convertidos para GitLab CI/CD, variáveis secretas, memberships, imagens de contêiner, runners e integrações no destino, conforme os responsáveis do plano.
- Comunicação aos usuários sobre o encerramento do uso dos repositórios no Azure DevOps (o pipeline não arquiva a origem Azure DevOps).

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	Projetos, repositórios e usuários do escopo, onda e destino de cada item e projetos fora de ondas; sem tokens, e-mails ou membros.	Markdown e CSV (artefato do pipeline)
Plano de ondas	Projetos do Azure DevOps de cada onda e destino de cada repositório, validado pelo pipeline.	Repositório do engajamento (YAML e CSV) e Markdown por onda
Relatório de usuários	Usuário no Azure DevOps × usuário no destino e situação (existe no destino, ausente, ambíguo ou sem e-mail), 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 por projeto (objetos lidos × importados) e relações com falha.	Markdown e JSON (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 e itens de tratamento manual.	Repositório do engajamento
Relatório de encerramento	Resultado das ondas, alertas aceitos, itens de tratamento manual, 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 *GitLab export* habilitada e as demais configurações do documento de pré-requisitos.
- O e-mail de cada usuário no Azure DevOps é igual ao e-mail do usuário no destino; quem não existir no destino no momento da importação fica com a autoria na conta da

importação, sem reatribuição depois.

- Repositórios da onda não recebem alterações no Azure DevOps durante a janela da onda.
- Depois da importação, o GitLab Congregate remove os membros diretos dos projetos importados (o pipeline não usa a opção de manter contribuidores); para as memberships, a matriz do GitLab Congregate orienta SAML (JIT), SCIM ou SAML Group Links no destino.
- O rollback só remove o que a onda criou dentro da janela acordada; usuários criados 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

- Repositórios TFVC (sem ferramenta da GitLab para migrar de TFVC para Git).
- Destino GitLab.com ou GitLab Dedicated (roadmap).
- Itens classificados como "não migra" na matriz do documento de pré-requisitos, como Azure Test Plans; 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 *GitLab export* habilitada e grupo pai privado ou interno.
- Host de migração dedicado com Docker, runner shell do engajamento e rede: host → Azure DevOps, destino e serviços da GitLab.
- PAT e usuário do Azure DevOps e token de administrador do destino, cadastrados como variáveis do projeto do engajamento.
- E-mails dos usuários iguais no Azure DevOps e no destino, e decisão entre criação pelo GitLab Congregate ou provisionamento por LDAP ou SAML antes das ondas.
- Aprovador de cada onda e do rollback, janelas e congelamento de escrita no Azure DevOps durante a janela.
- Validação funcional dos projetos críticos de cada onda e responsáveis pelos itens de tratamento manual.

Detalhamento no documento de pré-requisitos PR-MIG-03, enviado após a escolha do serviço.

11. Duração

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

A duração depende do volume de repositórios e pull requests, do número de projetos e de ondas, das janelas de mudança e da prontidão dos pré-requisitos. As referências consultadas não trazem estimativa oficial de duração para origem Azure DevOps. No pipeline, o job de migração de cada onda tem limite padrão de 8 horas. Toda estimativa de prazo depende do volume real, dos limites de API do Azure DevOps e da infraestrutura do destino, e não é garantia de prazo.

12. Documentos relacionados

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