

Documentação do Desafio de Automação

Objetivo: Avaliar a capacidade dos candidatos em projetar, construir e documentar fluxos de automação robustos e escaláveis utilizando uma plataforma low-code. O desafio aborda conceitos de lógica de programação, integração de sistemas e tratamento de erros.

Plataforma Low-Code: Streamline

Estamos considerando não só a implementação técnica dos fluxos, mas também a capacidade de resolução de problemas, o pensamento construtivo e lógico, e a qualidade da documentação apresentada. Isso significa que os candidatos deverão demonstrar clareza na abordagem, planejamento sistemático e detalhamento de suas estratégias, decisões e tratamento de erros, evidenciando uma solução robusta e escalável.

PARA SEGUIR COM ESSE DESAFIO, CONSIDERAMOS QUE VOCÊ JÁ TENHA FEITO O TUTORIAL DO STREAMLINE ANTERIORMENTE.

Solicitação de Reembolso

Visão Geral

Automatizar o processo de solicitação e reembolso de despesas de funcionários. O fluxo deve receber a solicitação, extrair dados do comprovante (OCR), validar informações, passar por aprovações e, se aprovado, integrar-se a um sistema para pagamento.

Para isso crie um projeto com nome **"TechLab - <Seu Nome>"**, coloque nos domínios sempre **"-<seunome>"**, por exemplo, **"reembolso-paulo-henrique"**.

1. Formulário de Solicitação de Reembolso

- **Campos Obrigatórios:**
 - Nome do solicitante
 - E-mail
 - CPF
 - Departamento

- Descrição da despesa
- Categoria da despesa (ex: Alimentação, Transporte, Hospedagem)
- Forma de pagamento (ex: Cartão de Crédito, Dinheiro)
- Upload do comprovante (imagem ou PDF)
- Banco
- Agência
- Conta

2. Fluxo de Processamento (Reembolso)

1. **Início:** O fluxo é iniciado pelo envio do formulário de reembolso.
2. **Extração de Dados - OCR (Nível 2+):**
 - Utiliza o bloco de API para chamar o serviço de OCR.
 - Extrai: valor, CNPJ/CPF do fornecedor e data.
3. **Aprovações (Nível 1+):**
 - Aprovação sequencial (Time RH, depois Financeiro).
 - Aprovação condicional (ex: valor exceder o valor de R\$200,00 aprovação do Sócio).
 - Notificações para os aprovadores via email caso seja recusado em alguma das etapas.
4. **Integrações com sistema de pagamentos (Nível 2+):**
 - **Sistema Financeiro:** Enviar dados para agendar o pagamento, que deverá ser simulado pelo disparo do webhook do discord. Contendo todos os dados da transação (adicione todos que você ache interessante).
5. **Finalização:**
 - Altera o status para "Aprovado" ou "Rejeitado".
 - Notifica o solicitante.

Documentação Técnica

Integração com OCR

- **Serviço:** Doc Extraction Tech4AI
- **Endpoint:** <https://api.tech4.ai/document/extract>

- **Método:** POST.
- **Autenticação:**

```
Unset
{
  Content-Type: application/json
  x-client-key: 5NxdN3rtEQUEW1o9SY3r1kSAS1we1HsW
}
```

- **Payload (Exemplo):**

```
Unset
{
  "file_url": "${activity.<id da variavel do seu form>.url}",
  "layout_id" : "f9ee4016-87e2-4717-b4dd-73cb8dcd503d"
}
```

- **Resposta (Exemplo):**

```
Unset
{
  "status": "ok",
  "extracted_data": {
    "cnpj": "11.189.345/0001-51",
    "data_documento": "2025-04-08",
    "total_pagamento": "184,82"
  }
}
```

- **Tratamento de Resposta:** Extrair os dados relevantes e armazenar no ticket.

Fases da Avaliação

Aqui está uma descrição mais detalhada de cada fase da avaliação, juntamente com os critérios específicos a serem observados:

1. Construção dos Fluxos (Peso: 60%)

- **Objetivo:** Avaliar a habilidade do candidato em *implementar* os fluxos na plataforma low-code, utilizando os blocos e recursos disponíveis de forma eficaz.
- **O que o candidato deve entregar:**
 - Os fluxos de reembolso completo e funcional na plataforma low-code.
 - Comentários *dentro* dos fluxos explicando a lógica e a função de cada bloco (ou grupo de blocos).
- **Critérios de Avaliação (para cada fluxo separadamente):**
 - **Implementação Correta:** O fluxo segue a lógica descrita na documentação preliminar (com possíveis ajustes justificados)?
 - **Uso Adequado dos Blocos:** O candidato utiliza os blocos corretos para cada tarefa (condicionais, loops, chamadas de API, etc.)?
 - **Lógica de Programação:** A lógica do fluxo está correta e eficiente?
 - **Tratamento de Erros:** O fluxo lida com erros de forma adequada (ex: falha no OCR, rejeição de aprovação)?
 - **Organização e Legibilidade:** O fluxo é fácil de entender e seguir? Os blocos estão organizados e nomeados de forma clara?
 - **Modularidade:** O candidato utilizou subfluxos ou componentes reutilizáveis para evitar duplicação?
 - **Integração com as ferramentas externas**

2. Testes (Peso: 40%)

- **Objetivo:** Avaliar a capacidade do candidato de *testar* os fluxos de forma completa e sistemática, garantindo que funcionem corretamente em diferentes cenários.
- **O que o candidato deve entregar:**
 - Um documento de testes (ex: planilha, documento de texto) descrevendo:
 - Os diferentes cenários de teste (ex: solicitação de reembolso com valor válido, falha no OCR, etc.).
 - Os dados de entrada utilizados em cada teste.

- Os resultados esperados de cada teste.
- Os resultados *reais* obtidos em cada teste.
- Evidências dos testes (ex: capturas de tela).

- **Critérios de Avaliação:**

- **Cobertura de Testes:** O candidato testou os fluxos com uma variedade suficiente de cenários (casos de sucesso, casos de erro, casos limite)?
- **Rigor dos Testes:** Os testes são detalhados e bem documentados?
- **Identificação de falhas:** O candidato identificou e corrigiu eventuais falhas nos fluxos durante os testes?
- **Documentação dos Resultados:** Os resultados dos testes são apresentados de forma clara e organizada?