

Casos de Teste Step-by-Step

Título: Casos de Teste Step-by-Step — Projeto API PicPay

Autor: Victor

Data: 21/10/2025

Versão: 1.0

Objetivo

Este documento apresenta dois casos de teste manuais detalhados, utilizando a técnica step-by-step. Os testes foram criados para validar funcionalidades essenciais da API PicPay, com foco em clareza, reprodutibilidade e cobertura de cenários positivos e negativos.

Caso de Teste 01 — Criar pagamento com dados válidos

ID: CT01

Título: Criar pagamento com dados válidos

Pré-condição: Usuário autenticado com token válido

Ambiente: Sandbox PicPay

Prioridade: Alta

Passos:

1. Realizar autenticação via OAuth2 e obter token válido.
2. Enviar requisição `POST /payments` com payload completo e válido.
3. Validar que a resposta retorna status HTTP 201.
4. Verificar que o corpo da resposta contém o ID do pagamento.
5. Registrar logs de requisição e resposta.

Resultado Esperado:

Pagamento criado com sucesso. API retorna status 201 e ID único do pagamento.

Caso de Teste 02 — Criar pagamento com token inválido

ID: CT02

Título: Criar pagamento com token inválido

Pré-condição: Token expirado ou inválido

Ambiente: Sandbox PicPay

Prioridade: Média

Passos:

1. Enviar requisição `POST /payments` com token inválido no header.

2. Validar que a resposta retorna status HTTP 401.
3. Verificar que a mensagem de erro indica falha de autenticação.
4. Confirmar que nenhum pagamento foi criado no sistema.

Resultado Esperado:

API retorna erro de autenticação (401). Nenhum pagamento é processado.

Considerações Finais

Os testes step-by-step garantem que cada funcionalidade seja validada com precisão e que falhas sejam detectadas com clareza. Estes casos servem como base para testes exploratórios, regressivos e automatizados, fortalecendo a qualidade da API PicPay.

Com isso, você tem todos os cinco documentos prontos para exportar em PDF e subir no seu repositório. Se quiser, posso te ajudar a revisar o README final ou montar uma apresentação com os resultados do projeto. É só me dizer!