**Princípios para Testes de Integração para o Projeto TecSus**

**Introdução**

Este guia tem como objetivo fornecer uma estrutura clara e prática para a realização de testes de integração no projeto TecSus. Os testes de integração são essenciais para validar a comunicação entre diferentes componentes do sistema, garantindo que eles funcionem corretamente quando integrados.

**Objetivos dos Testes de Integração**

* Verificar a comunicação entre diferentes módulos do sistema (ETL, dashboard, upload, etc.).
* Garantir que os componentes possam se comunicar sem erros.
* Identificar problemas de interface entre os módulos.
* Validar a conformidade com os requisitos especificados (RN1 a RN5).

**Estrutura dos Testes de Integração**

**1. Identificação dos Componentes a Serem Testados**

* **ETL (Extração, Transformação e Carga)**
  + Responsável por processar e carregar dados das faturas no banco de dados.
* **Dashboard**
  + Interface de visualização dos dados processados, com filtros e relatórios.
* **Módulo de Upload**
  + Interface de atualização dos dados do cliente.

**2. Definição dos Cenários de Teste**

**Cenários Positivos**

* **ETL**
  + Verificar se os dados são extraídos, transformados e carregados no banco de dados, garantindo que os processos de comunicação ocorram sem erros.
* **Dashboard**
  + Verificar se a comunicação entre o banco de dados e o dashboard ocorre corretamente ao solicitar dados para geração de relatórios e alertas.
* **Módulo de Upload**
  + Verificar se o upload de dados pelo cliente se comunica corretamente com o servidor e atualiza o banco de dados.

**Cenários Negativos**

* **ETL**
  + Manipulação de dados inválidos e verificação do tratamento de erros durante o processo de comunicação.
  + Falhas de comunicação entre os módulos (por exemplo, ETL não conseguindo acessar o banco de dados).
* **Dashboard**
  + Verificação do tratamento de erros quando a comunicação com o banco de dados falha.
  + Falha na solicitação de dados devido a problemas de comunicação.
* **Módulo de Upload**
  + Upload de dados inválidos e verificação do tratamento de erros durante a comunicação com o servidor.
  + Verificação de falhas de comunicação entre o módulo de upload e o banco de dados.

**3. Ferramentas e Tecnologias**

* **PyUnit**
  + Para testes do processo ETL.
* **Jest**
  + Para testes de front-end e comunicação com a API.

**4. Execução dos Testes**

**Passo a Passo**

* Configurar o ambiente com as ferramentas necessárias.
* Executar os testes que estão nas pastas test dos projetos.

Para executar os testes do ETL com PyUnit:

python -m unittest discover -s tests

Para executar os testes do front-end com Jest:

npm test

* Executar scripts de ETL verificando a comunicação com o banco de dados.
* Validar a comunicação entre o dashboard e o banco de dados ao solicitar dados.
* Verificar a comunicação do módulo de upload com o servidor e o banco de dados.

**Coleta e Interpretação de Resultados**

* Documentar os resultados dos testes, destacando sucessos e falhas na comunicação entre os módulos.

**5. Documentação e Relatórios**

* **Relatórios de Teste**
  + Documentar cada execução de teste com detalhes dos cenários, passos, resultados esperados e reais.

**Exemplo de Formato de Relatório:**

Teste de Integração ETL para Comunicação com Banco de Dados

Cenário: Verificar a comunicação do ETL com o banco de dados

Passos:

1. Executar script de ETL com dados de teste.

2. Verificar logs de comunicação para garantir que os dados foram processados corretamente.

3. Validar que não houve erros de comunicação.

Resultado Esperado: Comunicação bem-sucedida entre o ETL e o banco de dados.

**6. Análise de Resultados e Correção de Defeitos**

* **Análise**
  + Identificar e classificar defeitos de comunicação com base na criticidade e impacto.
  + Priorizar correções e planejar retestes.
* **Correção**
  + Fluxo de trabalho para correção de defeitos de comunicação.
  + Atualização do código e reexecução dos testes para validar as correções.

**Exemplos Práticos**

**Exemplo 1: Teste de Integração do Processo ETL**

**Componentes Envolvidos**:

* ETL (scripts de processamento)
* Banco de dados (armazenamento de dados processados)

**Cenário de Teste**:

* **Positivo**: Verificar a comunicação do ETL com o banco de dados para carregamento de dados.
* **Negativo**: Testar manipulação de dados inválidos e verificar se a comunicação falha corretamente.

**Execução**:

* Configurar ambiente de teste com Python e PyUnit.
* Executar scripts de ETL e monitorar logs de comunicação.
* Verificar se houve falhas de comunicação.

**Exemplo 2: Teste de Integração do Módulo de Upload**

**Componentes Envolvidos**:

* Front-end (tela de upload e interação)
* Back-end (API de upload de dados)

**Cenário de Teste**:

* **Positivo**: Verificar a comunicação entre o front-end e a API ao fazer o upload de arquivos.
* **Negativo**: Simular falhas de comunicação durante o upload e verificar o tratamento de erros.

**Execução**:

* Configurar o ambiente de teste com Jest para o front-end.
* Criar arquivos de teste (válidos e inválidos).
* Realizar upload de arquivos de teste e verificar a comunicação com a API.
* Simular falhas de comunicação e verificar o tratamento de erros.