## Documento de Cobertura de Testes

Por: Luis Felipe F. Silva

## ✅ Funcionalidades Prioritárias para Teste

### 🖥️ Back-End

No back-end, foram priorizadas duas funcionalidades principais:

* **Upload de Arquivos**
* **Download de Arquivos por Token**

Essas funcionalidades são o núcleo do sistema e representam os pontos mais sensíveis quanto à integridade de dados, validação e manipulação de arquivos.

## 🎯 Justificativa Crítica dos Testes Realizados

As funcionalidades de upload e download foram priorizadas porque representam:

* **O ponto de entrada e saída do sistema.**
* Áreas com alto risco de falha, especialmente em manipulação de arquivos binários e streams.
* Necessidade de garantir integridade — por isso o teste de download compara o conteúdo binário original com o retornado pela API.

Além disso:

* O upload foi testado com um arquivo real.
* O download foi testado por token, garantindo a segurança e funcionalidade do sistema sem expor o ID direto do banco.
* Casos de erro também foram considerados (upload sem arquivo).

Essa abordagem foca na **robustez da base do sistema**, validando que os arquivos são corretamente recebidos, armazenados e recuperados.

## ✅ Funcionalidades Prioritárias para Teste

### 🌐 Front-End

No front-end, foi implementado um conjunto completo de **testes end-to-end com Cypress**, cobrindo todo o fluxo de interação do usuário:

* Upload de arquivo
* Renomear arquivo
* Excluir arquivo
* Acionar download

## 🎯 Justificativa Crítica dos Testes Realizados

No front, o foco foi em testes e2e com Cypress porque:

* Testam a aplicação como um **usuário real** (via UI e ações reais).
* Validam não só a integração com a API, mas também a **interface, estados e comportamento da tela.**
* Cobrem todo o ciclo de vida de um arquivo: **upload → listagem → renomear → download → excluir.**

A abordagem de e2e foi escolhida por:

* Ser a forma mais eficaz de garantir que **todas as partes do sistema (frontend + backend) estão integradas corretamente.**
* Permitir verificação visual e funcional do fluxo de uso.

Além disso, os testes foram escritos de forma a:

* **Limpar os arquivos após execução**, mantendo o sistema limpo.
* Serem **reutilizáveis e confiáveis**, sem interferência entre eles.