

# Documento Visão do Projeto

**Nuvem - Sistema de Armazenamento Seguro de Arquivos**

**Aluno:** Everton Hian dos Santos Pinheiro

# 1. Objetivo

Este documento tem como objetivo definir as necessidades de alto nível, características gerais e requisitos do sistema **Nuvem**, um serviço de armazenamento em nuvem seguro com foco acadêmico. O projeto servirá como base para demonstrar boas práticas de:

- **Segurança da informação** (autenticação forte, criptografia)
- **Arquitetura de software** (API RESTful, modularidade)
- **Gerenciamento de dados** (armazenamento escalável, backups)

## 2. Descrição do Projeto

**Nuvem** é um sistema de armazenamento de arquivos que permite:

- Upload/download de arquivos **públicos** (compartilháveis via link) ou **privados**
- Autenticação em duas etapas (2FA) via e-mail
- Criptografia de nomes de arquivos e dados sensíveis
- Gerenciamento de pastas e quota de armazenamento
- Logs detalhados para auditoria

**Tecnologias Principais:**

- **Backend:** Python (Flask), SQLAlchemy, JWT
- **Banco de Dados:** PostgreSQL
- **Segurança:** HTTPS, bcrypt, Fernet (AES-128)
- **Infraestrutura:** Docker (opcional), WebSocket para notificações

## 3. Justificativa e Escopo

### 3.1 Justificativa

Projeto acadêmico para demonstrar:

- Implementação de **políticas de segurança** (GDPR-like)
- Padrões de **autenticação moderna** (JWT + 2FA)

### 3.2 Escopo

**Incluído:**

- Cadastro/login com 2FA
- Upload/download de arquivos

- Compartilhamento via link
- Criptografia de metadados
- Sistema de logs e backups

**Excluído:**

- Edição colaborativa de arquivos (ex.: Google Docs)
- Sincronização em tempo real entre dispositivos

## 5. Requisitos

### 5.1 Funcionais

ID	Descrição	Ator
RF01	Cadastrar usuário com 2FA	Usuário
RF02	Upload de arquivos (públicos/privados)	Usuário
RF03	Gerenciar pastas (criar/renomear/excluir)	Usuário
RF04	Compartilhar arquivos via link	Usuário
RF05	Compartilhar pastas	Usuário
RF06	Excluir Conta	Usuário
EF07	Alterar Senha	Usuário

### 5.2 Não Funcionais

ID	Descrição
RNF01	Criptografia AES-128 para nomes de arquivos
RNF02	Autenticação JWT com expiração
RNF03	Logs de todas as operações sensíveis
RNF04	Verificação de dois fatores
RNF05	HTTPS em comunicação

## 6. Restrições Tecnológicas

- **Backend:** Flask (Python 3.10+)
- **Frontend:** Não incluso neste projeto (API-first)
- **Banco de Dados:** PostgreSQL 14+
- **Segurança:**
  - HTTPS obrigatório
  - Senhas com hash bcrypt
  - Tokens JWT expiráveis

## **7. Critérios de Aceitação**

1. Usuários conseguem se cadastrar e ativar 2FA
2. Arquivos privados não são acessíveis sem autenticação
3. Links de compartilhamento expiram conforme configurado
4. Logs registram IP, horário e ação do usuário