

[illegible]

Responsáveis

1. Equipe

Coordenador

Fernando Luiz Lopes Carvalho

Desenvolvimento

Fernando Luiz Lopes Carvalho

Documento de Visão do Projeto: X-DataLink

1. Nome do Projeto

X-DataLink

2. Contexto e Justificativa

Colocar em prática os conceitos estudados na disciplina de Sistemas Distribuído. O projeto visa criar um sistema distribuído que permita aos usuários armazenar, compartilhar e acessar arquivos. Ele utilizará autenticação social (Google) para facilitar o acesso e proverá uma interface intuitiva para interação com os arquivos.

3. Objetivos do Projeto

Objetivo Geral

Desenvolver um sistema distribuído de troca e armazenamento de arquivos entre usuários, com foco em segurança, escalabilidade e simplicidade.

Objetivos Específicos

1. Implementar uma API em Node.js para conectar o banco de dados a página web.
 2. Gerenciar o armazenamento dos arquivos usando serviços de nuvem ou armazenamento local.
 3. Prover autenticação social via Google para facilitar o acesso dos usuários.
 4. Oferecer funcionalidades como upload, download, compartilhamento e controle de permissões de acesso.
 5. Registrar logs de atividades para auditoria.
-

4. Escopo do Projeto

Incluído no Escopo:

- Cadastro e autenticação de usuários via autenticação social.
- Upload, download e exclusão de arquivos.
- Compartilhamento de arquivos com controle de permissões (leitura/escrita).
- Interface web para gerenciamento dos arquivos e compartilhamentos.

- Registro de logs de atividades dos usuários.

Fora do Escopo:

- Criação de aplicativos móveis(*Em análise*).
 - Integração com outros sistemas além do Google Auth.
-

5. Descrição dos Usuários

Usuários Finais:

- Usuários que necessitam armazenar e compartilhar arquivos com outros.
 - Perfis variados: profissionais, estudantes e equipes colaborativas.
-

6. Requisitos Funcionais e Não Funcionais

Requisitos Funcionais

- **RF01:** Permitir login via autenticação social do Google.
- **RF02:** Permitir upload de arquivos com limite de tamanho configurável.
- **RF03:** Prover download de arquivos por seus proprietários e usuários autorizados.
- **RF04:** Permitir compartilhamento de arquivos com permissões específicas.
- **RF05:** Registrar logs de atividades, incluindo uploads, downloads, compartilhamentos e exclusões.

Requisitos Não Funcionais

- **RNF01:** A API será construída em Node.js.
 - **RNF02:** O banco de dados deve suportar escalabilidade.
 - **RNF03:** O sistema deve ser seguro, utilizando HTTPS para comunicação.
 - **RNF04:** Garantir uma interface web responsiva e acessível.
 - **RNF05:** Os arquivos devem ser armazenados em serviços como AWS S3, Google Cloud Storage ou equivalentes.
-

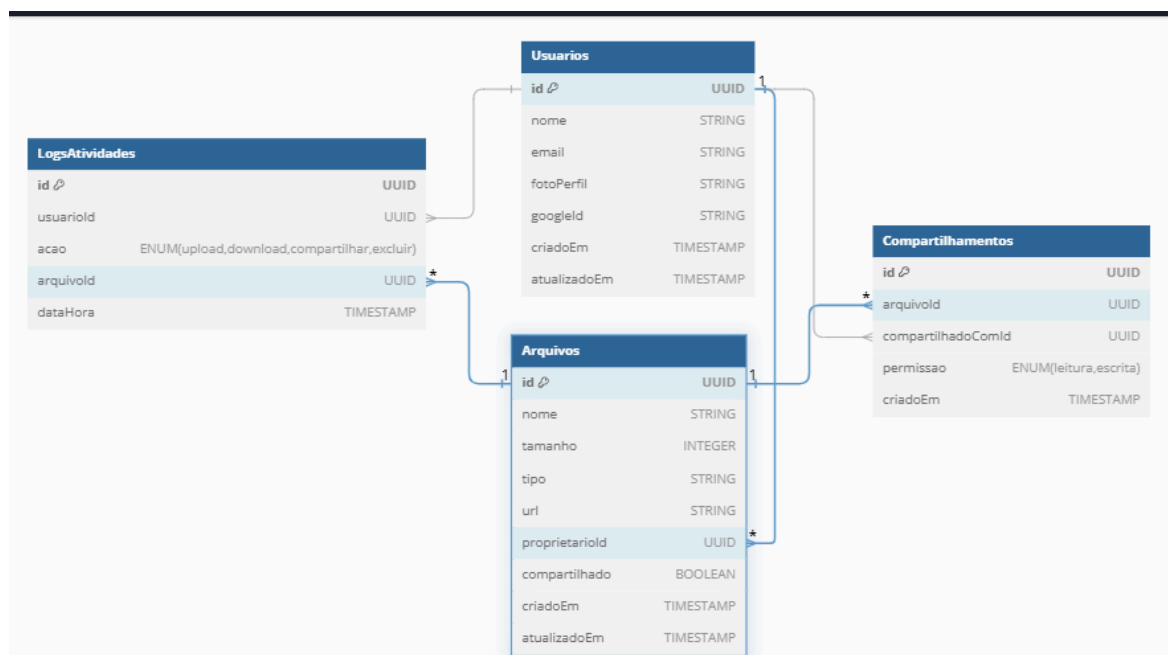
7. Modelo de Dados

- **Entidades:**
 - Usuários
 - Arquivos
 - Compartilhamentos

- Logs de Atividades

- **Relacionamentos:**

- Usuários possuem arquivos.
- Arquivos podem ser compartilhados com outros usuários.
- Logs registram ações de usuários relacionadas a arquivos.



8. Arquitetura do Sistema

- **Back-end:**

- Node.js com Express.
- Prisma ORM para gerenciamento do banco de dados.

- **Banco de Dados:**

- Relacional (PostgreSQL).

- **Front-end:**

- Aplicação web integrada com a API do back-end.

- **Armazenamento de Arquivos:**

- Soluções em nuvem (AWS S3) e armazenamento local.

- **Autenticação:**

- OAuth 2.0 com Google.

9. Riscos e Mitigações

| Risco | Mitigação |
|-------------------------------------|--|
| Falha na integração com Google Auth | Testar previamente com diferentes contas e cenários. |
| Crescimento inesperado de usuários | Utilizar arquitetura escalável e serviços em nuvem. |
| Perda de arquivos armazenados | Implementar backups automáticos no sistema. |

10. Cronograma

1. Planejamento: 2 semanas
 - Definição do escopo e requisitos.
2. Desenvolvimento da API: 4 semanas
 - Configuração do banco de dados e endpoints principais.
3. Integração com Front-end: 4 semanas
 - Criação da interface web e integração com a API.
4. Testes e Ajustes Finais: 2 semanas
 - Testes de usabilidade, carga e segurança.
5. Lançamento: 1 semana.

| TAREFA/SEMANA | SEMANA 1 | SEMANA 2 | SEMANA 3 | SEMANA 4 | SEMANA 5 | SEMANA 6 | SEMANA 7 | SEMANA 8 |
|--------------------------|----------|----------|----------|----------|----------|----------|----------|----------|
| PLANEJAMENTO | X | X | | | | | | |
| DESENVOLVIMENTO DA API | | | X | X | X | X | | |
| INTEGRAÇÃO COM FRONT-END | | | | X | X | X | X | |
| TESTES E AJUSTES FINAIS | | | | | | | X | X |