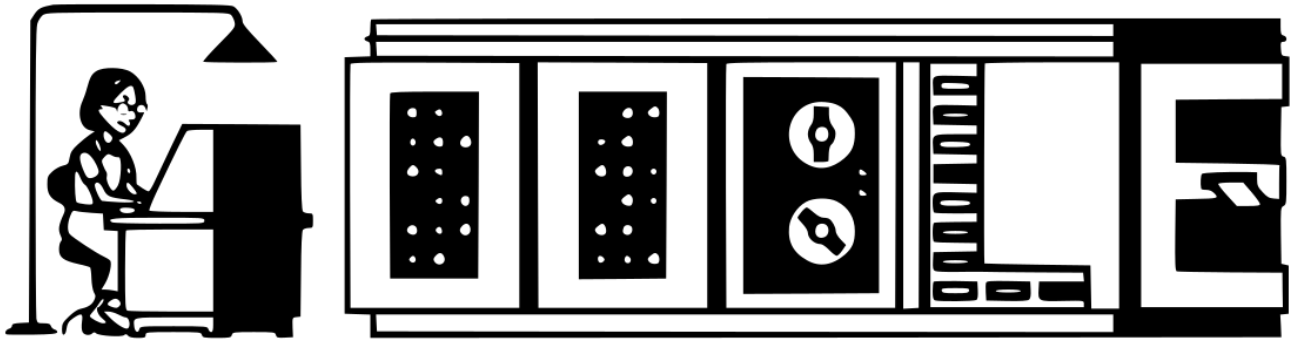


Hopper



- **Título do Projeto:** Sistema de Gerenciamento e Distribuição de Relatórios Dinâmicos com Power BI Embedded.
- **Nome do Estudante:** Gabriel Deglmann Kasten.
- **Curso:** Engenharia de Software.
- **Data de Entrega:** [Data].

Resumo

Este projeto visa desenvolver uma plataforma integrada para gerenciar processos de ETL (Extract, Transform, Load), relatórios dinâmicos no Power BI e distribuí-los de forma segura em uma aplicação web. O sistema utiliza Airflow para orquestração de pipelines de dados, Power BI Embedded para disponibilização de dashboards e autenticação baseada em roles (RBAC) e RLS (Row-Level Security) para controle de acesso. Os principais pontos abordados são a integração de tecnologias de BI com sistemas web, segurança de dados e gerenciamento de workflows.

1. Introdução

Organizações modernas demandam relatórios atualizados e personalizados para tomada de decisão. Entretanto, processos manuais de ETL e a distribuição não centralizada de relatórios geram gargalos de eficiência e riscos de segurança. Este projeto propõe uma solução automatizada para esses desafios.

A integração de ETL, ferramentas de BI e controle de acesso granular é essencial para a engenharia de software, envolvendo arquitetura de sistemas, segurança da informação e gestão de dados.

Tendo como principal objetivo desenvolver um sistema que facilite o gerenciamento e distribuição de relatórios com controle de acesso e segurança, se faz essencial os seguintes pontos:

1. Implementar RLS para filtragem de dados conforme perfil do usuário.
2. Criar uma interface web intuitiva para visualização de dashboards.

- 3. Garantir escalabilidade do processo de ETL.

2. Descrição do Projeto

O Hopper é um Sistema de gerenciamento de dados e relatórios com Power BI Embedded, autenticação customizada e pipelines ETL automatizados. O sistema visa mitigar processos manuais de extração e transformação de dados, falta de centralização no acesso a relatórios e garantir um controle adequado de permissões e visibilidade de dados.

Tendo em vista o tempo e conhecimento técnico, o sistema não abordará análise de dados em tempo real e não incluirá desenvolvimento de visualizações personalizadas fora do Power BI.

3. Especificação Técnica

Descrição detalhada da proposta, incluindo requisitos de software, protocolos, algoritmos, procedimentos, formatos de dados, etc.

3.1. Requisitos de Software

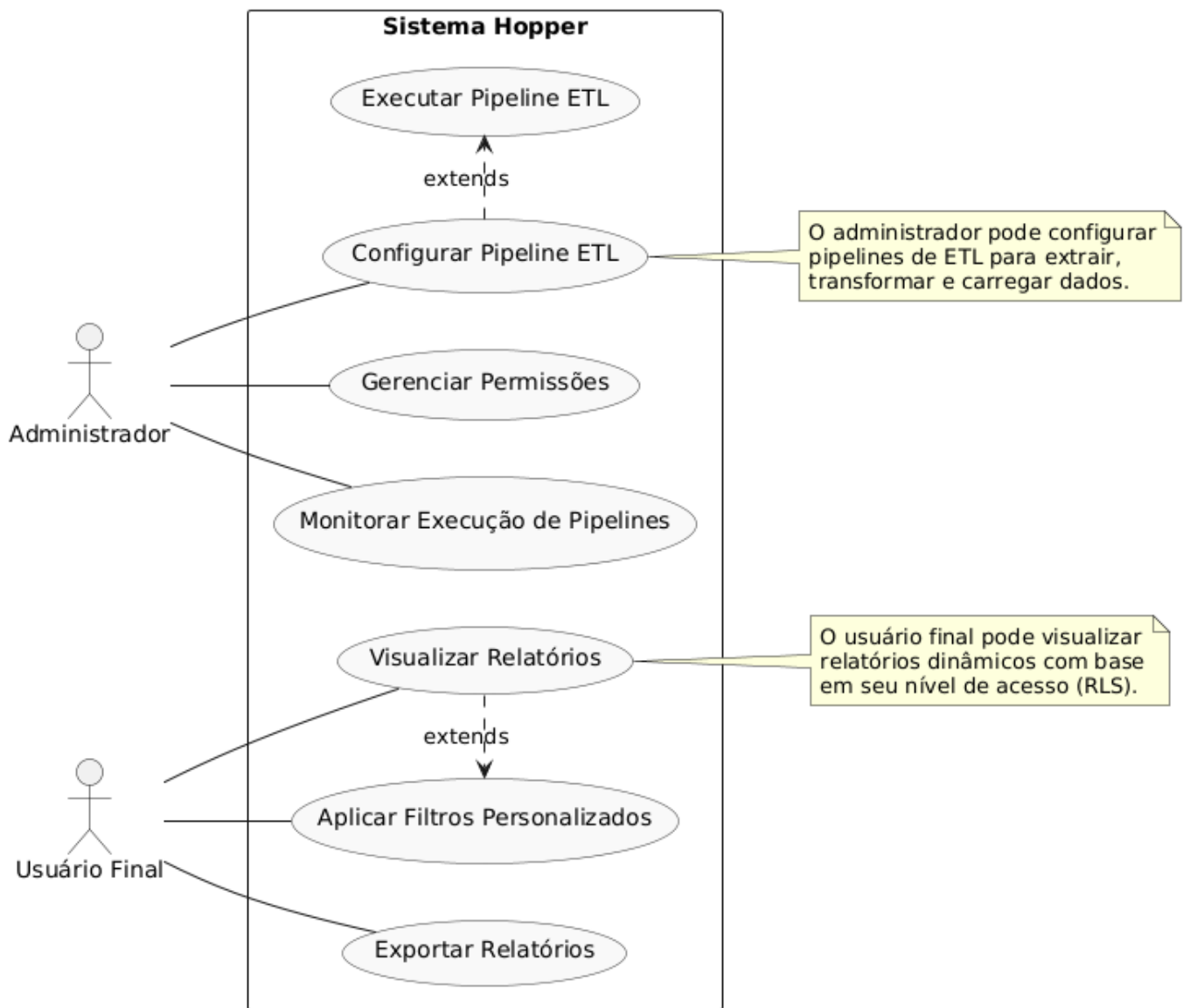
- **Lista de Requisitos Funcionais:**

- RF01: Execução automatizada de pipelines ETL com Airflow.
- RF02: Geração de relatórios no Power BI com atualização programada.
- RF03: Autenticação de usuários via sistema RBAC.
- RF04: Aplicação de RLS conforme perfil do usuário.
- RF05: Publicação de dashboards em uma aplicação web via Power BI Embedded.

- **Lista de Requisitos Não Funcionais:**

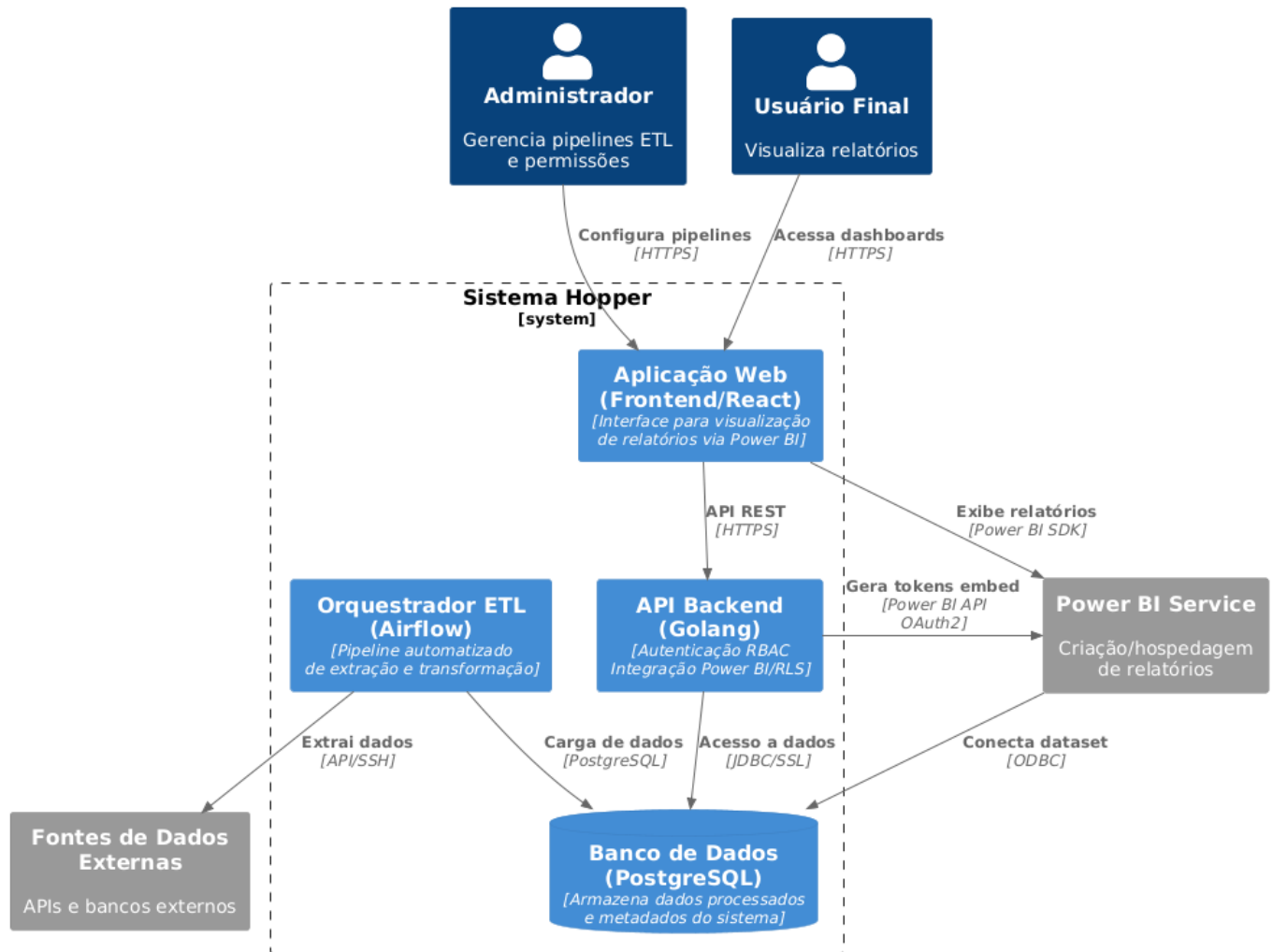
- RNF01: Tempo máximo de 5 minutos para execução do ETL.
- RNF02: Disponibilidade de 99,9% para a aplicação web.
- RNF03: Criptografia de dados em trânsito e repouso.

- **Representação dos Requisitos:**

Diagrama de Casos de Uso - Hopper**3.2. Considerações de Design****• Visão Inicial da Arquitetura:**

- Camada ETL: Airflow para orquestração.
- Camada de Dados: Banco de dados PostgreSQL e modelagem.
- Camada de BI: Power BI para visualização dos dashboards.
- Camada Web: ReactJS (frontend) e Golang (backend).

• Padrões de Arquitetura: MVC na camada web e Microserviços para ETL.**• Modelo C4:**



3.3. Stack Tecnológica

- **Linguagens de Programação:**

- Python (ETL),
- JavaScript/React (frontend).
- Golang (backend)

- **Frameworks e Bibliotecas:**

- Apache Airflow,
- React,
- Gin (API)
- Power BI REST API.
- Pandas
- Numpy
- Psycpg2

- **Ferramentas de Desenvolvimento e Gestão de Projeto:**

- Docker (containerização)

- Git (versionamento)
- Render (Deploy)

3.4. Considerações de Segurança

Analisando possíveis questões de segurança e como mitigá-las, foi decidido que como medida mínima de contenção, é necessário que os seguintes requisitos devem ser atendidos:

- Autenticação via OAuth2.
- RLS no Power BI para restrição de dados.
- Auditoria de logs de acesso.

4. Próximos Passos

Após aprovação do documento, os próximos passos são em ordem:

1. Criação de um backlog.
2. Implementação de um contêiner com Airflow e pipelines funcionais com diferentes peculiaridades.
3. Desenvolvimento de relatórios no Powerbi utilizando dos dados importados.
4. Desenvolvimento da interface web com integração ao Power BI Embedded.
5. Implementação de funcionalidades para ativação e desativação de cargas.
6. Permitir o gerenciamento facilitado das cargas, painéis, workspaces, grupos, RLS, etc.
7. Testes de segurança e carga.

5. Referências

Listagem das fontes de pesquisa, frameworks, bibliotecas e ferramentas que serão utilizadas.

- [Airflow Docs](#)
- [PowerBI API](#)
- [WSTG](#)
- [Data Pipelines with Apache Airflow \(Livro\)](#)
- [Just the Docs \(Documentação Geral\)](#)
- [Plantuml \(Diagramas\)](#)
- [Docker](#)
- [React](#)
- [Pandas](#)
- [Numpy](#)
- [Psycpg2](#)
- [Gin Gonic](#)
- [Render](#)
- [Git](#)
- [Python](#)
- [Golang](#)
- [Powerbi](#)
- [Powerbi Embedded](#)
- [Biblioteca de ícones](#)

Para controle de metas e entregas foi utilizado o [Trello](#).

6. Avaliações de Professores

- Considerações Professor/a:

- Considerações Professor/a:

- Considerações Professor/a: