



- **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

Em ambientes corporativos que lidam com grandes volumes de dados, equipes de negócios enfrentam atrasos e inconsistências ao consolidar e analisar informações manualmente. O Hopper surge para resolver esse problema, oferecendo uma plataforma que automatiza o fluxo de ETL, centralizando e padronizando a preparação de dados e disponibilizando dashboards interativos via Power BI Embedded, garantindo relatórios sempre atualizados, seguros e escaláveis para suportar decisões estratégicas.

1. Introdução

- **Contexto:** Organizações modernas demandam relatórios precisos e personalizados para apoiar a tomada de decisão. No entanto, processos manuais de ETL e a distribuição descentralizada de relatórios geram gargalos de eficiência, atrasos na entrega de insights e riscos de segurança.
- **Justificativa:** O Hopper mitiga esses desafios ao orquestrar pipelines de dados com Apache Airflow, automatizando a extração, transformação e carga de informações, e centralizando dashboards no Power BI Embedded. Dessa forma, o sistema reduz erros manuais, acelera a disponibilização de insights e fortalece a governança de dados.
- **Objetivos:** O objetivo principal é oferecer uma plataforma que automatiza o fluxo de ETL e disponibiliza dashboards interativos. Objetivos secundários incluem implementar controle de acesso granular, monitorar métricas de execução e permitir o gerenciamento de relatórios.

2. Descrição do Projeto

- **Tema do Projeto:** Desenvolvimento de uma plataforma de gerenciamento e distribuição de relatórios dinâmicos, chamada Hopper, que automatiza o processo de ETL e utiliza Power BI Embedded para visualização de dados.
- **Problemas a Resolver:**
 - Atrasos e inconsistências na consolidação e análise manual de dados.
 - Gargalos de eficiência em processos de ETL.
 - Riscos de segurança na distribuição de relatórios.
 - Falta de relatórios atualizados e escaláveis para decisões estratégicas.

- **Limitações:** O projeto não abordará a criação de visualizações customizadas além das capacidades do Power BI, nem o desenvolvimento de algoritmos de machine learning para análise preditiva. A gestão de infraestrutura de hardware subjacente fica a cargo dos serviços de nuvem (Render.com).

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:**

- **Requisitos Funcionais (RF):**

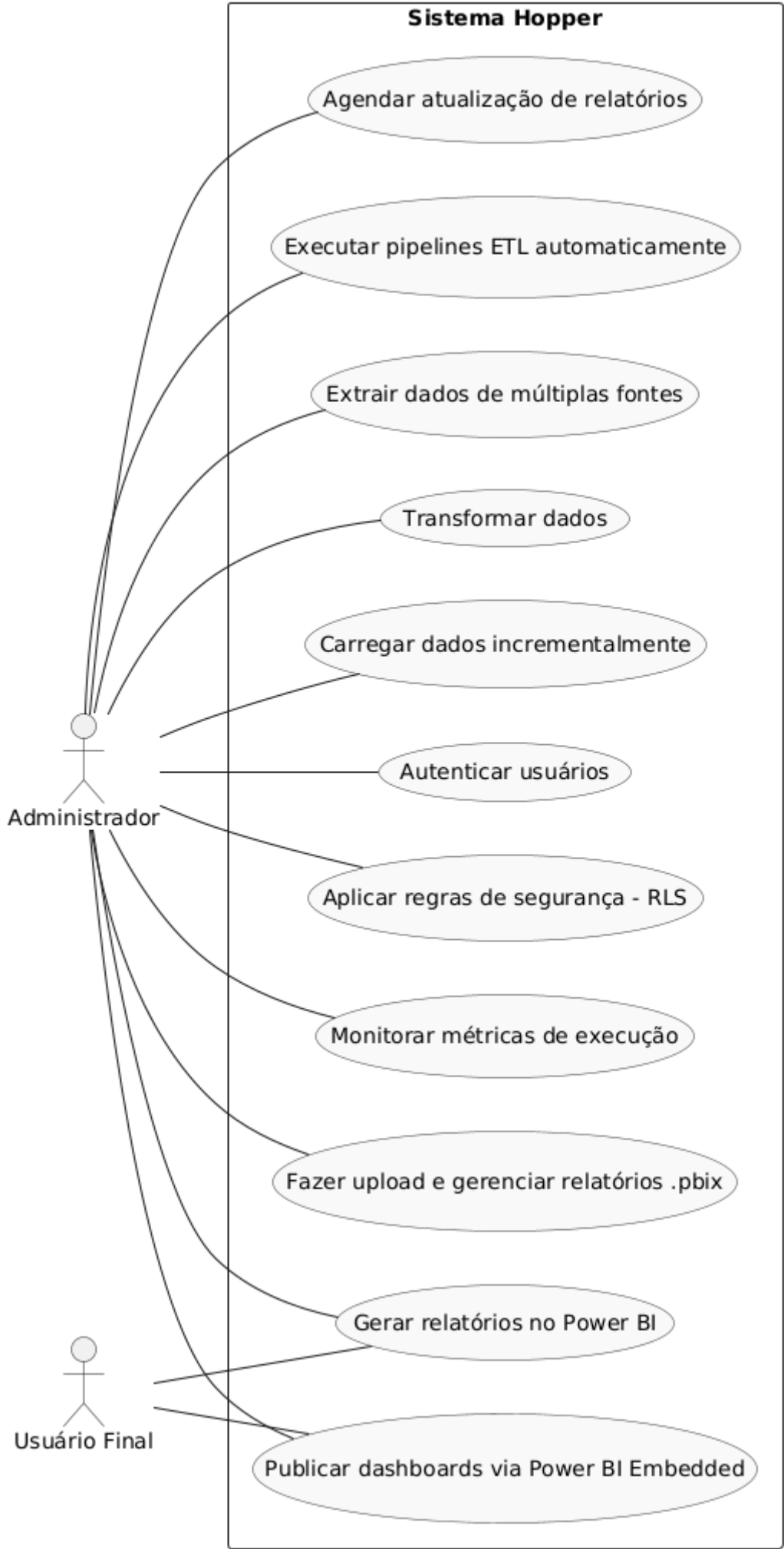
- **RF01:** O sistema deve permitir a execução automatizada de pipelines ETL com Apache Airflow.
- **RF02:** O sistema deve extrair dados de múltiplas fontes (CSV, APIs REST, PostgreSQL).
- **RF03:** O sistema deve transformar dados (limpeza, enriquecimento, agregação) com Python (Pandas/Numpy).
- **RF04:** O sistema deve realizar carga incremental no banco PostgreSQL.
- **RF05:** O sistema deve gerar relatórios no Power BI importando datasets do PostgreSQL.
- **RF06:** O sistema deve permitir autenticação de usuários via OAuth2 (Google/Microsoft).
- **RF07:** O sistema deve aplicar regras de segurança em nível de linha (RLS) nos relatórios.
- **RF08:** O sistema deve publicar dashboards do Power BI em uma aplicação Web React.
- **RF09:** O sistema deve monitorar métricas de execução do Airflow.
- **RF10:** O sistema deve permitir upload e gerenciamento de relatórios .pbix via Power BI REST API.
- **RF11:** O sistema deve permitir atualização programada dos relatórios no Power BI.

- **Requisitos Não Funcionais (RNF):**

- **RNF01:** O sistema deve definir cargos (Admin, Analista, Visualizador) e integrar com provedores OAuth2.
- **RNF02:** O sistema deve garantir tempo máximo de execução de 5 minutos para os pipelines.
- **RNF03:** O sistema deve criptografar dados em trânsito (TLS 1.3) e em repouso (AES-256).
- **RNF04:** O sistema deve ser implantado com auto-scaling horizontal e balanceamento de carga.
- **RNF05:** O sistema deve gerar logs de auditoria centralizados.
- **RNF06:** O sistema deve utilizar operadores customizados em Python para ETL.
- **RNF07:** O sistema deve integrar-se à Power BI REST API para automação.
- **RNF08:** O sistema deve utilizar tokens JWT no backend para autorização.
- **RNF09:** O sistema deve utilizar a biblioteca powerbi-client-react no frontend para embed seguro.

Representação dos Requisitos:

Diagrama de Casos de Uso - Hopper



3.2. Considerações de Design

- **Visão Inicial da Arquitetura:**
 - **Camada ETL:** Airflow foi escolhido por sua robustez e flexibilidade na orquestração de pipelines de dados, facilitando automação e monitoramento.
 - **Camada de Dados:** PostgreSQL oferece confiabilidade, escalabilidade e suporte avançado à modelagem de dados, sendo ideal para centralizar informações. Além de já vir como metabase da plataforma do airflow.
 - **Camada de BI:** Power BI permite visualizações interativas e integração nativa com múltiplas fontes, acelerando a entrega de insights.
 - **Camada Web:** ReactJS e FastAPI proporcionam desenvolvimento ágil, interfaces modernas e APIs performáticas, facilitando integração e manutenção.
- **Padrões de Arquitetura:** O padrão MVC na camada web separa responsabilidades, melhorando organização e testabilidade. Microserviços no ETL aumentam escalabilidade, isolamento de falhas e flexibilidade na evolução dos pipelines.
- **Modelos C4:**

Diagrama de Contexto

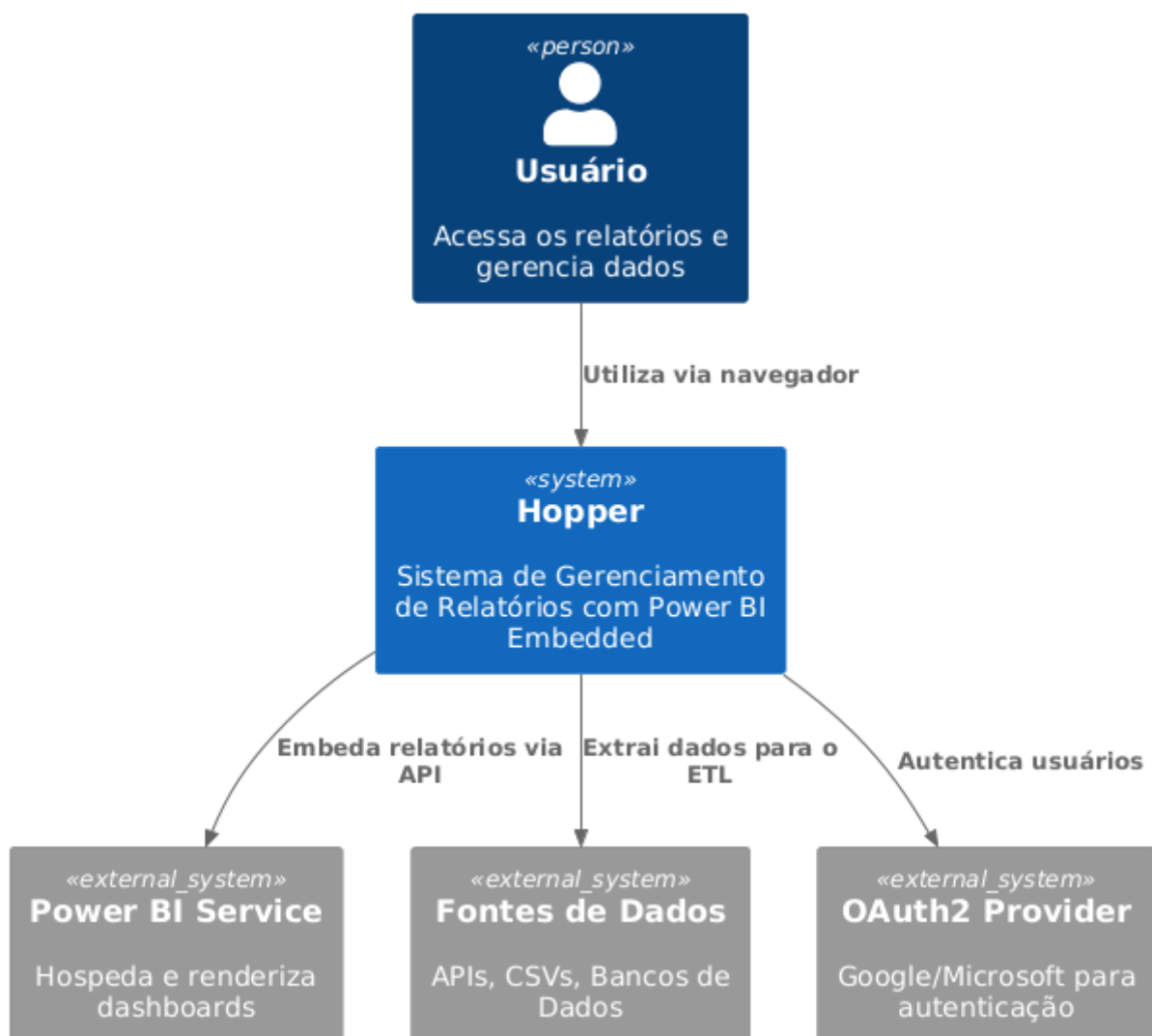


Diagrama de Container

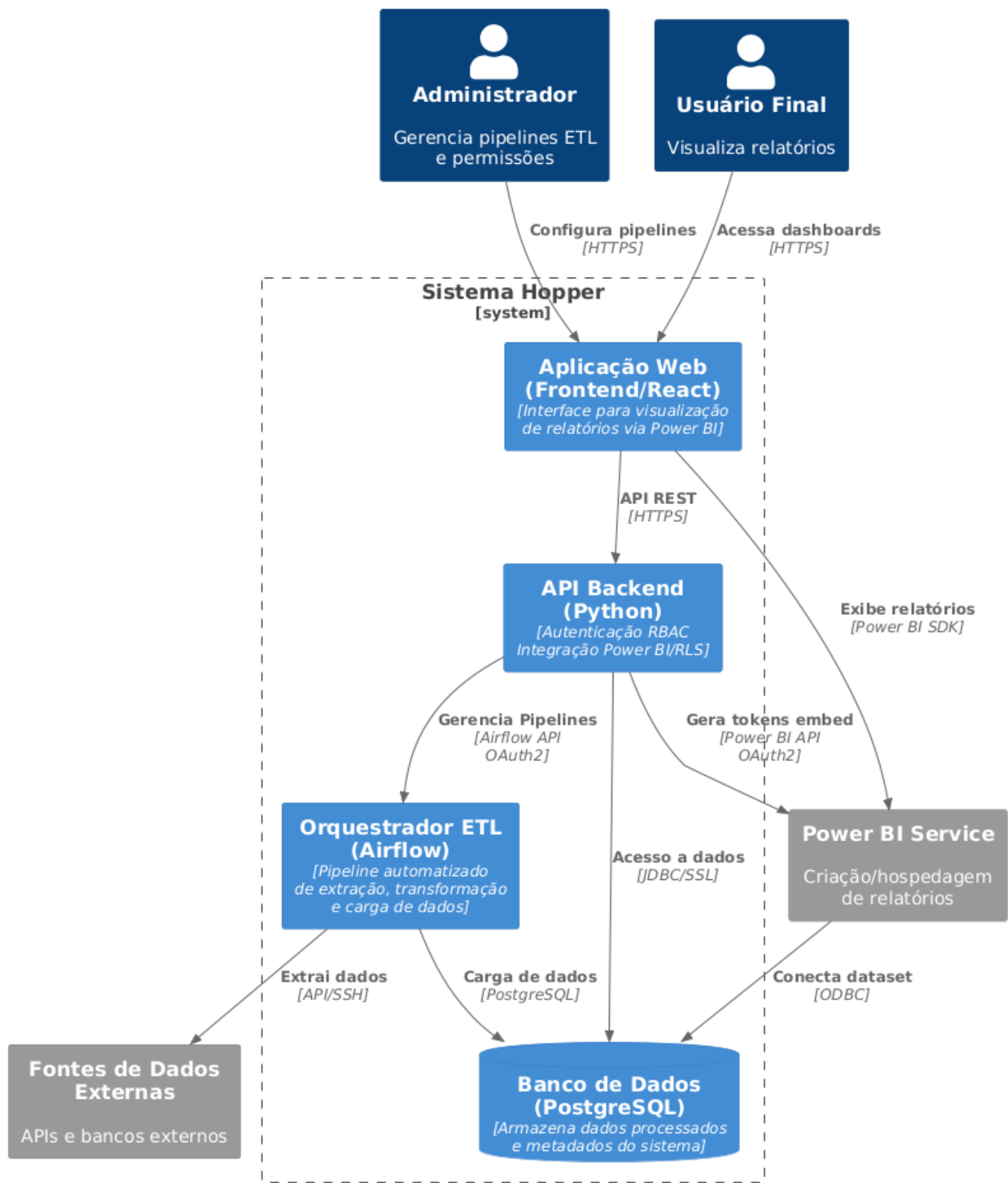
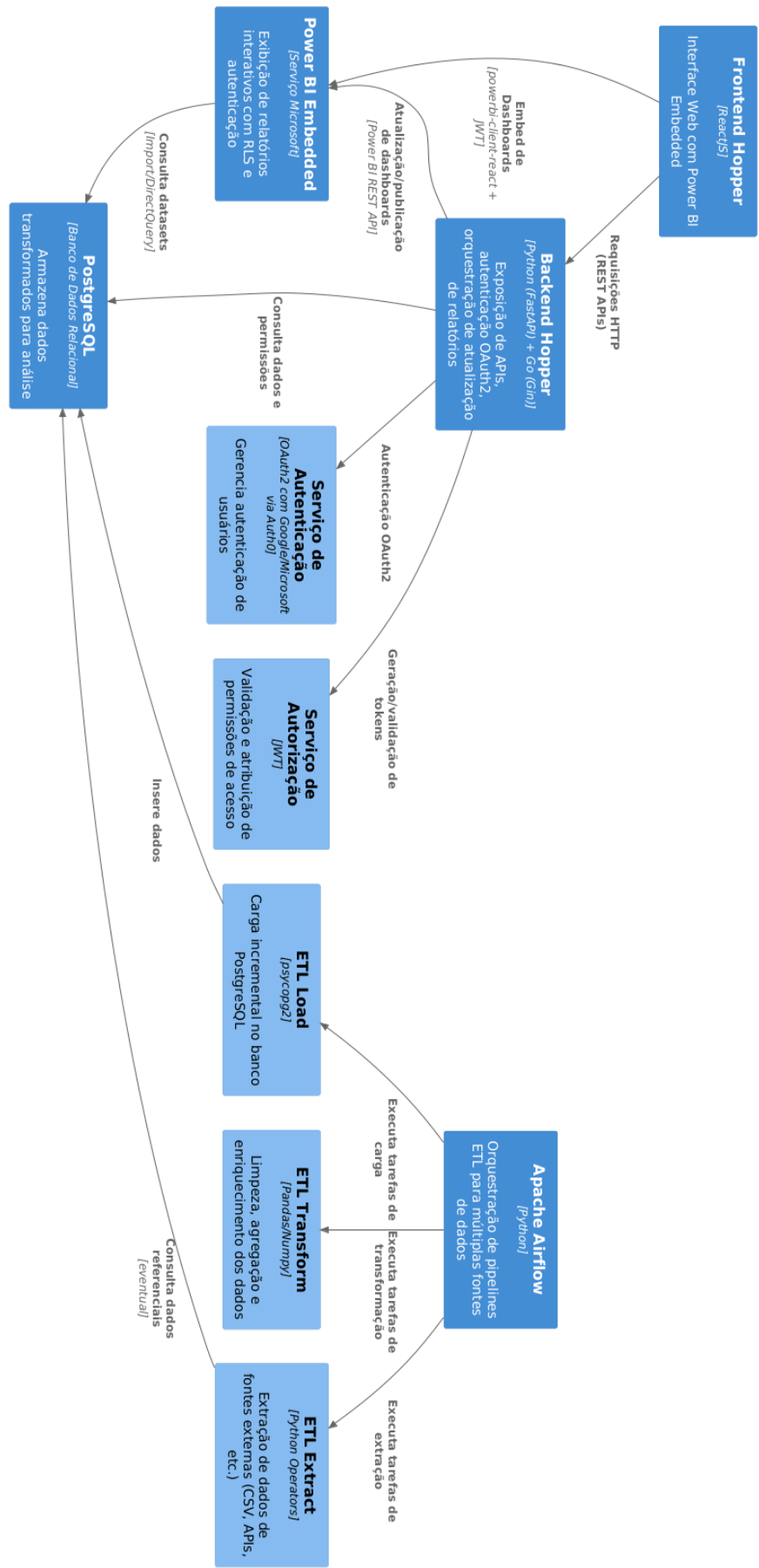


Diagrama de Componentes



3.3. Stack Tecnológica

3.3. Stack Tecnológica

- **Linguagens de Programação:**

- Python (ETL e backend via FastAPI): Escolhido pela robustez em manipulação de dados, ampla comunidade e integração nativa com Airflow e bibliotecas de ciência de dados.
- JavaScript/React (frontend): Permite interfaces modernas, responsivas e integração facilitada com Power BI Embedded.

- **Frameworks e Bibliotecas:**

- Apache Airflow: Referência em orquestração de pipelines ETL, com escalabilidade e monitoramento avançados.
- React: Framework consolidado para construção de SPAs, garantindo experiência de usuário fluida.
- FastAPI: Framework web rápido e eficiente, com suporte a APIs REST e validação automática de dados.
- Power BI REST API: Necessária para automação e gerenciamento de relatórios Power BI.
- Pandas e Numpy: Padrão para processamento e análise de dados em Python.
- Psycopg2: Driver robusto para integração entre Python e PostgreSQL.

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

- Docker: Facilita a containerização, portabilidade e replicação do ambiente.
- Git: Controle de versionamento confiável e colaborativo.
- Render: Plataforma de deploy automatizado e escalável.
- Trello: Organização visual e ágil das tarefas do projeto.
- Figma: Design colaborativo e prototipação rápida de interfaces.

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:

- **Controle de Acesso e Autenticação**

- *Questão:* Usuários não autorizados podem tentar acessar dados ou relatórios restritos.
- *Mitigação:* Implementação de autenticação via OAuth2 (Google/Microsoft), garantindo que apenas usuários autenticados possam acessar o sistema. Exemplo: ao tentar acessar um dashboard, o usuário é redirecionado para o provedor OAuth2 e só recebe acesso após validação.

- **Restrição de Dados Sensíveis**

- *Questão:* Usuários autenticados podem visualizar dados além do seu escopo de permissão.

- *Mitigação:* Aplicação de Row-Level Security (RLS) nos relatórios Power BI, restringindo a visualização de dados conforme o perfil do usuário. Exemplo: um analista de uma filial só visualiza dados da sua unidade.

- **Auditoria e Rastreamento de Acesso**

- *Questão:* Falta de rastreabilidade dificulta a identificação de acessos indevidos ou anomalias.
- *Mitigação:* Geração e centralização de logs de acesso e ações dos usuários, permitindo auditoria detalhada. Exemplo: cada login, visualização ou download de relatório é registrado com timestamp e identificação do usuário.

- **Exposição de APIs e Integrações**

- *Questão:* APIs expostas podem ser alvo de ataques (ex: brute force, injection).
- *Mitigação:* Uso de tokens JWT para autorização nas APIs e limitação de requisições (rate limiting).

- **Gerenciamento de Relatórios e Uploads**

- *Questão:* Upload de arquivos .pbix maliciosos pode comprometer a plataforma.
- *Mitigação:* Validação de arquivos no upload, restrição de tipos permitidos e verificação de integridade.

Essas medidas, combinadas, visam garantir confidencialidade, integridade e disponibilidade dos dados e relatórios gerenciados pela plataforma Hopper.

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.
3. Desenvolvimento de relatórios no Power BI.
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, etc.

5. Referências

- [Airflow Docs](#)
- [PowerBI API](#)
- [WSTG](#)
- [Data Pipelines with Apache Airflow \(Livro\)](#)
- [Just the Docs \(Documentação Geral\)](#)
- [Plantuml \(Diagramas\)](#)
- [Docker](#)
- [React](#)
- [Pandas](#)
- [Numpy](#)
- [Psycpg2](#)
- [Render](#)

- [Git](#)
- [Python](#)
- [FastAPI](#)
- [Powerbi](#)
- [Powerbi Embedded](#)
- [Powerbi React](#)
- [Biblioteca de ícones](#)
- [Trello](#)
- [Figma](#)

6. Apêndices

Principais Termos

- **Power BI Embedded:** Serviço da Microsoft que permite incorporar dashboards e relatórios interativos do Power BI em aplicações web, proporcionando visualização de dados segura e customizável para usuários finais.
- **ETL (Extract, Transform, Load):** Processo de extração, transformação e carga de dados, fundamental para integração e preparação de informações provenientes de múltiplas fontes.
- **Apache Airflow:** Plataforma open source para orquestração de workflows, utilizada para automatizar e monitorar pipelines de dados.
- **Row-Level Security (RLS):** Mecanismo de segurança que restringe o acesso a linhas específicas de dados em relatórios, conforme o perfil do usuário.
- **Power BI REST API:** Interface de programação que permite automação e gerenciamento de recursos do Power BI, como relatórios e workspaces.
- **Render:** Plataforma de cloud para deploy automatizado de aplicações, com suporte a autoescalabilidade e balanceamento de carga.

7. Avaliações de Professores

- Considerações Professor/a:
- Considerações Professor/a:
- Considerações Professor/a: