

# Bruno Chiconato

Contato: [E-mail](#) | (42) 9 9997-9114 | Curitiba/PR

Perfil: [LinkedIn](#) | [Portfólio](#) | [GitHub](#)

---

## Resumo Profissional

Engenheiro de Dados com experiência no desenvolvimento de software para integração de dados e construção de pipelines **ETL/ELT** escaláveis. Sólidos conhecimentos em **Python**, **Spark** e **SQL**, com experiência prática na orquestração de jobs com **Airflow** e na arquitetura de soluções em ambientes cloud (**AWS** e **Databricks**). Focado em aplicar boas práticas de engenharia, como **CI/CD** e Infraestrutura como Código (**Terraform**), para construir e manter plataformas de dados robustas que conectam e capacitam o negócio.

---

## Competências Técnicas

- **Stack Principal:** Python, Spark/PySpark, SQL, AWS, Databricks, Airflow, Git, CI/CD (GitHub Actions), Docker
  - **Cloud/Data Warehousing:** AWS (S3, EC2, Lambda, Glue, Redshift, Athena, IAM, Kinesis Firehose), Snowflake, BigQuery
  - **Orquestração e ETL/ELT:** Airflow, dbt, Jenkins, Pentaho, Apache Hop
  - **Big Data e Arquitetura:** Delta Lake/Unity Catalog, Data Lake, Data Warehouse, Data Lakehouse
  - **Bancos de Dados:** PostgreSQL, SQL Server, MySQL
  - **DevOps e IaC:** Terraform, Docker, Git/GitHub, GitHub Actions
  - **Idiomas:** Inglês nível profissional
- 

## Experiência Profissional

### Engenheiro de Dados — Insight Wave | 08/2024 – 08/2025

- **Garanti 100% de integridade e confiabilidade** nos dados de gestão, **reduzindo o tempo de atualização de dashboards em 96%** (de 20 min para 40s), ao desenvolver e **orquestrar com Jenkins um pipeline de ETL em Python** que substituiu um conector de BI instável, consumindo a API do Jira e estruturando os dados na camada Bronze de um **Data Lake**.
- **Elevei a qualidade e a manutenibilidade do código** do pipeline, atingindo **85% de cobertura de testes**, através da implementação de um ciclo de desenvolvimento profissional com Git/GitHub, **testes unitários/integração com pytest** e **deploy automatizado via esteiras de CI/CD com GitHub Actions**.
- **Automatizei 100%** do processo de coleta e tratamento de dados de emplacamentos, **liberando a equipe de ~40 horas/mês de trabalho manual** e **reduzindo o tempo de disponibilização da informação em 98%** (de 2h para 2 min), ao construir um **pipeline de web scraping em Python** com lógica de **carga incremental**.
- **Aumentei a estabilidade e a performance** de um job de processamento de NFe, alcançando uma **redução de 90% no tempo de execução** e **eliminando falhas críticas** de sistema, ao **refatorar a arquitetura da query T-SQL**, substituindo a query legada por uma estrutura mais eficiente.
- **Liberei 50% da janela de processamento** de dados, **reduzindo o tempo de execução de um ETL crítico** de 26 para 13 minutos, ao reescrever e **otimizar o código SQL** legado com a implementação de **CTEs e técnicas de query pushdown**.
- **Reduzi a contenção de recursos e o custo computacional** no banco de dados, diminuindo as **leituras lógicas em 80%** e o **tempo de execução de queries em 42%**, através da otimização estratégica com a **criação de índices**.
- **Aumentei a confiança dos stakeholders** nos dados, **eliminando inconsistências e duplicidades** em dashboards críticos, ao desenvolver um **processo de padronização** que unificou dados entre o ERP TOTVS e fontes XML, aplicando **regras de negócio e ajustes de modelagem**.
- **Habilitei análises de dados escaláveis e confiáveis**, garantindo a **governança e a qualidade** desde a ingestão, ao **projetar e implementar a arquitetura de um Data Lake** com **camadas Bronze e Gold**, metadados de rastreabilidade e rotinas de **carga incremental automatizadas** com Power Automate e **Apache Hop**.

### Estagiário de Dados — Renault Group | 05/2023 – 07/2024

- **Garanti 100% de precisão** na base de dados SG3, desenvolvendo **scripts de validação** e rotinas de padronização com **SQL** para identificar e corrigir inconsistências de forma contínua.
- **Liberei 10 horas de trabalho manual por semana** (~40 horas/mês) da equipe ao automatizar integralmente a geração de relatórios operacionais, orquestrando fluxos com **Power Automate** e desenvolvendo macros avançadas em **VBA**.
- **Melhorei a acessibilidade e o uso da informação**, construindo fluxos automatizados com **Power Query** para coleta de dados e desenvolvendo relatórios interativos com tabelas dinâmicas e gráficos em **Excel**, culminando no **treinamento dos usuários finais** para garantir a adoção das ferramentas.
- **Estruturei o processo de governança de dados** para a base SG3, mapeando fontes primárias, estabelecendo um dicionário de dados e trabalhando com a equipe de TI para validar e documentar todas as regras de negócio implementadas.

## Projetos

### ScholarStream – Ingestão de Dados Acadêmicos em Baixa Latência com AWS e Snowflake | [Repositório](#)

- Implementei um pipeline de dados **near real-time** para ingestão da API OpenAlex, utilizando **Python** e **AWS Kinesis Firehose**.
  - Estruturei os dados no **Snowflake**, com uma camada **RAW** em **VARIANT** e views na camada **CURATED** para consumo analítico.
  - Configurei **Role-Based Access Control (RBAC)** e políticas de mascaramento de dados para garantir a **segurança**.
  - Provisionei toda a infraestrutura de forma automatizada com **Terraform** e configurei **CI/CD** com **GitHub Actions**.
- Tecnologias:** Python, AWS (Kinesis, Lambda, S3), Snowflake, Terraform, GitHub Actions, Streamlit.

### arXiv Lakehouse – Arquitetura Moderna com Delta Lake, Databricks e AWS | [Repositório](#)

- Construí um **pipeline de dados** completo na plataforma **Databricks**, ingerindo dados da API do arXiv e processando-os através das camadas **Bronze**, **Silver** e **Gold**.
  - Gerenciei a arquitetura **Lakehouse** com **Delta Lake** em **S3** e governança de dados com **Unity Catalog**.
  - Empacotei a lógica de transformação com **PySpark** em formato wheel para jobs serverless e defini data contracts com **Pydantic**.
- Tecnologias:** Databricks, PySpark, Delta Lake, Unity Catalog, AWS S3, Pydantic, GitHub Actions.

### StockPulse – Plataforma Analítica de Dados Financeiros com Airflow e Spark | [Repositório](#)

- Orquestrei um **pipeline diário** com **Apache Airflow** para ingestão de dados da API do Yahoo Finance.
  - Implementei um processo de **ETL** com **Spark** para transformar os dados brutos (JSON) em formato analítico (CSV) e carregá-los no **PostgreSQL**.
  - Configurei a infraestrutura completa em containers **Docker**, incluindo o cluster **Spark** (master/worker).
- Tecnologias:** Apache Airflow, Spark, Python, Docker, PostgreSQL, MinIO, Metabase.

### Northwind Analytics – DataOps com dbt, AWS e Terraform | [Repositório](#)

- Desenvolvi um **pipeline** de transformação de dados utilizando **dbt**, estruturado em camadas (staging, intermediate, marts) para análises de finanças e marketing.
  - Automatizei o **deploy** e os **testes** (dbt run, dbt test) em um ambiente **AWS ECS Fargate** através de esteiras de **CI/CD** com **GitHub Actions**.
  - Provisionei toda a infraestrutura cloud (**ECR**, **ECS**, **RDS**, **S3**, **IAM**) como código usando **Terraform**.
- Tecnologias:** dbt, AWS (ECS Fargate, ECR, RDS, S3), Terraform, GitHub Actions, Docker, PostgreSQL.

---

## Formação Acadêmica

- Pós-Graduação em Engenharia de Dados e Big Data — PUC Minas | Conclusão: 2026
- Bacharelado em Engenharia Elétrica — UTFPR | Conclusão: 2024

---

## Certificações

- **Astronomer Certification for Apache Airflow 3 Fundamentals** — Astronomer
- **Databricks Accredited Databricks Fundamentals** — Databricks
- **AWS Certified Cloud Practitioner** — Amazon Web Services (AWS)