Gustavo Ferreira de Oliveira

Node Insight

SÃO PAULO
2025

Cataloging in Publication
Library and Documentation Service
Institute of Technology and Leadership (INTELLI)
Data entered by the author.
(Cataloging record with international cataloging data, according to NBR 14724. The record will be completed later, after approval and before the final version is deposited. The completion of the cataloging record is the responsibility of the institution's library.)

Sobrenome, Nome

Título do trabalho: subtítulo / Nome Sobrenome do autor; Nome e Sobrenome do orientador. – São Paulo, 2025.

nº de páginas : il.

Trabalho de Conclusão de Curso (Graduação) – Curso de [Ciência da Computação] [Engenharia de Software] [Engenharia de Hardware] [Sistema de Informação] / Instituto de Tecnologia e Liderança.

Bibliografia

1. [Assunto A]. 2. [Assunto B]. 3. [Assunto C].

CDD. 23. ed.

Acknowledgments

I would like to express my sincere gratitude to my family and friends, who supported me throughout this entire journey. I would also like to especially thank Marcelo Santana, Fernando Frascino, and Guilherme Cestari (my academic advisor) for their invaluable support during the development and validation of the project.

Resumo

Ferreira de Oliveira, Gustavo. **Node Insight**. 2025. 33. TCC (Graduação) – Curso de Engenharia de Computação, Instituto de Tecnologia e Liderança, São Paulo, 2025.

Este trabalho apresenta o desenvolvimento do Node Insight, uma solução otimizada para o BTG Empresas voltada à correlação de dados de eventos de clientes por meio de estruturas de grafos. O objeto de estudo concentra-se nos desafios de fragmentação de dados e ineficiência operacional no atendimento de pequenas e médias empresas. O objetivo principal é consolidar as interações dos usuários no Portal e Aplicativo BTG Empresas em um banco de dados Neo4j, permitindo uma visão integrada e automatizada das jornadas dos clientes. A metodologia adotada fundamenta-se no desenvolvimento ágil em cinco *sprints*, abrangendo desde a análise de viabilidade financeira até a implementação de uma API GraphQL sob os princípios da *Clean Architecture* e testes de performance com o Grafana K6. Como resultados, o projeto entrega uma infraestrutura escalável capaz de reduzir o estouro de SLAs e fornecer *insights* estratégicos sobre o ciclo de vida dos produtos. Conclui-se que a migração para um modelo relacional de grafos não apenas otimiza o diagnóstico de problemas pelo time de suporte, mas também fortalece a tomada de decisão baseada em dados, garantindo maior estabilidade e eficiência para o ecossistema bancário.

Palavras-Chave: banco de dados em grafos; neo4j; btg empresas; api graphql; eficiência operacional.

ABSTRACT

[Ferreira de Oliveira, Name. **Title of the work**. 2025. 33. Final course project (Bachelor) – Course Computer Engineering, Institute of Technology and Leadership, São Paulo, 2025.]

This study presents the development of Node Insight, an optimized solution for BTG Empresas designed to correlate customer event data through graph structures. The object of study focuses on the challenges of data fragmentation and operational inefficiency in serving small and medium-sized enterprises. The main objective is to consolidate user interactions from the BTG Empresas Portal and App into a Neo4j database, enabling an integrated and automated view of customer journeys. The methodology is based on agile development across five sprints, ranging from financial feasibility analysis to the implementation of a GraphQL API under Clean Architecture principles and performance testing with Grafana K6. As a result, the project delivers a scalable infrastructure capable of reducing SLA breaches and providing strategic insights into product lifecycles. It is concluded that the migration to a graph relational model not only optimizes problem diagnosis for the support team but also strengthens data-driven decision-making, ensuring greater stability and efficiency for the banking ecosystem.

Key words: graph databases; neo4j; btg empresas; graphql api; operational efficiency.

List of Illustrations

Figure 1 – Architecture Diagram.....18.
Figure 2 – Server running with API response.....20.
Figure 3 – Example of pagination usage.....20.
Figure 4 – List of all active products belonging to all clients….....21.
Figure 5 – List of all deactivated products belonging to all clients.….....22.
Figure 6 – A list of all active products of a client, filtered by CNPJ/CPF.....22.
Figure 7 – All CI/CD pipelines developed.….....23.

Summary

1 Introduction 8
1.1. Partner Company Context 8
1.2. Problem Definition (Corporate Pain Point) 8
1.3. Proposed Solution and Expected Contribution 9
1.4. Business Objectives 10
1.5. Structure of the thesis/dissertation 10
2 Solution Development 11
2.1 Applied Rationale 11
2.1.1 Business Area Rationale 12
2.1.2 Technological rationale for the solution 13
2.1.3 Fundamentals of Management and Development Methods 14
2.2 Specification and Development 15
2.2.1 Requirements and Specifications 15
2.2.2 Architecture and Technology 17
2.2.2.1 Integration into the IT Ecosystem 18
2.2.3 Development and Implementation (MVP) 18
2.2.4 Testing and Technical Evaluation 20
2.2.4.1 Data Validation and Mock Testing 21
2.2.4.2 Code Quality and CI/CD Integration 22
2.2.4.3 Performance Benchmarking (Grafana K6) 23
2.2.4.4 User Acceptance and Support Integration 24
2.3 Assessment of Impact and Contribution to the Business 25
2.3.1 Defining Corporate Success Metrics 25
2.3.1.1 Measurement Methodology 25
2.3.2 Results and Impact Analysis 26
2.3.3 Cost-Benefit Analysis 27
2.3.3.1 Financial Analysis 27
2.3.3.2 Infrastructure Cost Reduction Strategy 28
2.3.4 Critical Success Factors and Lessons Learned 29
2.3.4.1 Critical Success Factors 30
2.3.4.2 Lessons Learned 30
3 Conclusion 31
References 32
Annexes 33

1 Introduction

In the current financial landscape, service agility and data analysis precision have become critical competitive differentiators for banking institutions. Despite its recent history and exponential growth in the small and medium enterprise (SME) segment, BTG Empresas faces the typical challenges of rapid expansion: data fragmentation and system overhead during peak periods. Currently, the decentralization of information across different databases compromises the rapid diagnosis of operational issues and limits the business team's ability to generate personalized strategic insights.

In this context, Node Insight emerges as a project aimed at transforming the platform's data management through graph-oriented database technology. The relevance of this work lies in its ability to correlate, in real-time, customer actions within the BTG Empresas Portal and App, offering a holistic and centralized view of the user journey. The primary objective is to develop a scalable solution that not only reduces technical support response times (SLA) but also provides intelligent dashboards for strategic decision-making. By implementing a resilient and integrated architecture, the project seeks to raise the standard of operational efficiency and ensure the stability of the banking ecosystem even under high demand.

1.1. Partner Company Context

BTG Empresas is the arm of BTG Pactual focused on the Small and Medium Enterprises (SMEs) and Corporate ecosystem. With only five years of operation, the unit has established itself as a reference in the banking and credit sector in Latin America, exceeding the mark of 30,000 active clients (the current MVP focus is on 20,000 predominantly SME clients). The area affected by the project includes the Customer Service, Operations, and Business Intelligence departments, which are responsible for managing the BTG Empresas Portal and App.

Despite rapid growth, the operation faces infrastructure and data challenges-such as data fragmentation and decentralized databases-that may limit scalability.

1.2. Problem Definition (Corporate Pain Point)

Currently, at BTG Empresas, each team within the area maintains its own database containing information about the companies that have used its services.

The support team, responsible for assisting customers and resolving issues, must be familiar with available products and services, as well as new releases. Furthermore, they must have access to and familiarity with each team's various databases to handle the tickets received through the service channel.

For example, in the case of an unconfirmed Pix transaction, support agents must:

I. Access Database 1 to verify if the transaction contract was created and signed by the authorized parties.
II. Consult Database 2, which contains the transaction validation records.
III. Check Database 3, where the transmission of the transaction to the service responsible for processing with the Central Bank is recorded.

If the agent cannot determine what happened during the operation after this preliminary analysis, the issue is escalated to the responsible development team. This manual and decentralized process often results in SLA (Service Level Agreement) breaches.

On the other hand, the business team lacks a dedicated tool to track the lifecycle of new projects or analyze results over time. This hinders the creation of specific product strategies or personalized offerings for clients on the BTG Empresas Portal and App. When specific product data needs to be visualized, each respective team develops simple, non-standardized dashboards in Metabase, resulting in a lack of a centralized solution.

Finally, not all teams in the area have auto-scaling tools for their products or services, which leads to daily integration challenges. This scenario is aggravated during periods of high demand when there is an overload in the usage flow of the portal and the app. This occurs on critical dates for high-volume payment processing, such as FGTS, IPVA, and payroll, among others.

1.3. Proposed Solution and Expected Contribution

The primary objective of Node Insight is to develop an optimized solution that correlates customer action event data from the BTG Empresas Portal and App with their respective companies using a graph-oriented approach. By implementing a Neo4j database structured with nodes and edges, the project transitions away from fragmented and decentralized bases toward a unified relational model that enhances

the visualization of complex customer interactions. This technical core is supported by a GraphQL API built on Clean Architecture principles to ensure modularity and scalability, while automated data ingestion via AWS SNS and Lambda functions facilitates real-time processing.

This solution contributes directly to the bank's operational efficiency by providing an interactive dashboard that accelerates problem diagnosis for the support team and eliminates manual queries across multiple databases, effectively reducing SLA breaches. Simultaneously, Node Insight empowers the business team with strategic insights into product lifecycles and customer behavior patterns, while its architecture incorporates auto-scaling and performance optimization to handle critical high-demand periods like payroll processing, ensuring standardized visualization and long-term technical resilience.

1.4. Business Objectives

This will enable a more integrated and automated view of customer interactions, reducing data fragmentation and improving operational efficiency.

Additionally, an interactive dashboard will be developed for analyzing this data, serving two main audiences:

A. **Support Team:** Will have a tool that speeds up problem diagnosis, avoiding manual queries across multiple databases and reducing SLA breaches.
B. **Business Team:** Will be able to track the product life cycle, identify customer behavior patterns, and generate strategic insights for developing new solutions and service personalization.
C. **Area Developers:** Will be able to publish data on product usage by their clients following the defined data standard, resulting in a unified database of event and customer data.

1.5. Structure of the thesis/dissertation

The following section describes the structural organization of this thesis, outlining the content of the two subsequent chapters that detail the development, validation, and final considerations of the Node Insight project.

The second chapter, titled Solution Development, constitutes the technical and analytical core of this work, presenting the creation of the platform across four integrated perspectives. From a technological standpoint, it details the implementation of a native graph-oriented database using Neo4j and a GraphQL API governed by the principles of Clean Architecture to ensure system modularity and independence from external frameworks. Regarding the business context, the chapter explains how the solution addresses the needs of the BTG Empresas SME and Corporate ecosystem by unifying fragmented data to provide a "360-degree view" of customer interactions. The impact on the company is analyzed through the lens of operational efficiency, specifically the reduction of SLA breaches and the optimization of infrastructure costs, which are projected to decrease by 25% to 30% through the use of ARM-based AWS architectures. Finally, the chapter presents the results derived from technical evaluations, including performance benchmarking with Grafana k6 and usability testing with the support team, which led to the refinement of the NeoDash and Metabase dashboards.

The third and final chapter, Conclusion, synthesizes the findings of the research and confirms the achievement of the project's initial objectives. It discusses the broader business impacts, validating the high return on investment evidenced by a projected profit of approximately R$ 3.9 million. This section provides strategic recommendations for the future evolution of Node Insight, highlighting its inherent scalability and its potential as a foundation for Artificial Intelligence (AI) applications such as fraud analysis and churn monitoring. Furthermore, the conclusion outlines the knowledge transfer plan, detailing the technical documentation accessible via Swagger and the GitHub repository intended to ensure the continuity and maintenance of the solution by the company's internal development teams.

2 Solution Development

2.1 Applied Rationale

The solution is justified by its direct impact on the operational bottleneck of BTG Empresas, transforming how data flows from the system to the end-user to solve real-world business friction.

- **Transformation of the Support Workflow:** Instead of forcing support agents (L1 and L2) to manually navigate through multiple decentralized databases to investigate a single "Pix" failure, the solution applies a relational intelligence model. By presenting data through a unified graph interface, we eliminate the "manual search nightmare," directly addressing the human context of deadline pressure and reducing the frequency of SLA breaches.
- **Persona-Based Visualization Strategy:** The choice to integrate both NeoDash and Metabase is a methodological response to the diverse technical backgrounds within the bank. This "Applied Rationale" ensures that while technical analysts have the depth to explore complex node relationships, non-technical business managers have an accessible, table-oriented environment to track product life cycles and behavioral patterns without a steep learning curve.
- **Economic Sustainability and Scalability:** From a business perspective, the technical choices are applied to maximize ROI (Return on Investment). By leveraging ARM-based architectures and reserved cloud instances, the project justifies itself through a projected 25% to 30% reduction in infrastructure costs, proving that a more advanced technical solution can also be more cost-effective for the bank's long-term growth.

2.1.1 Business Area Rationale

To ensure the solution addresses the operational complexities of the company, it is essential to align the project with both the specific concepts of the financial sector and global market best practices for data visualization. During the analysis phase, a comprehensive benchmark of dashboard and graph visualization tools was conducted to identify the most suitable platform for handling complex relational data. The evaluation focused on three primary open-source solutions:

Tool	Key Characteristics	Best Use Case
Metabase	A low-code environment with an intuitive interface ideal for rapid prototyping and general data	Day-to-day operational analysis and

	exploration across multiple sources 10 .	table-oriented tracking for non-technical users.
NeoDash	Specifically designed for the **Neo4j ecosystem**, it utilizes libraries like @react-force-graph to directly expose relationships between nodes and edges.	Projects where visualizing interconnected graph structures is the central requirement.
Apache Superset	An enterprise-grade solution offering flexible queries, varied visualization options, and robust data governance for large datasets.	Complex projects requiring dashboards that aggregate highly diverse data sources.

Beyond off-the-shelf tools, the benchmark identified that using TypeScript-based libraries (such as D3.js, Plotly.js, and Highcharts) remains a high-flexibility alternative. These libraries allow for full control over interactivity and visual style when integrating custom dashboards directly into front-end frameworks.

The evaluation highlights that NeoDash is the superior choice for working within the Neo4j environment due to its native graph rendering. However, to meet the needs of all user profiles at BTG Empresas, a hybrid approach utilizing NeoDash for deep relational analysis and Metabase for accessible operational metrics was adopted to balance technical power with organizational usability.

2.1.2 Technological rationale for the solution

The technological foundation of Node Insight is built upon a cohesive integration of graph-based data modeling, modern architectural patterns, and serverless infrastructure, specifically selected to address the unique challenges of the BTG Empresas ecosystem. At the core of the solution lies Neo4j, a native graph database that represents information through nodes and edges rather than traditional tables. This choice is technically justified by the project's primary need to correlate complex customer interaction events-such as logins, payments, and product

usage-where traditional relational databases like PostgreSQL would suffer from inefficient and complex query performance when traversing multiple relationship layers. By utilizing the Cypher query language, the system can extract relational insights with much greater agility and lower overhead than standard SQL.

To ensure long-term maintainability and system stability, the application is developed using Clean Architecture and a GraphQL API. This architectural choice is critical because it decouples the core business logic from external frameworks and database drivers, allowing the platform to evolve or integrate with new tools without compromising the stability of its underlying rules. This is complemented by a Serverless Architecture utilizing AWS SNS and Lambda functions for data ingestion. This approach is technically justified by the frequent demand spikes encountered at BTG Empresas during critical high-volume payment windows-such as FGTS, IPVA, and payroll processing where automated processing and auto-scaling capabilities are essential to prevent integration failures and ensure high availability.

Finally, the analytical layer provides a balanced ecosystem through the integration of NeoDash and Metabase. While NeoDash serves the technical need for deep graph exploration and relationship visualization, Metabase was incorporated to support daily operational tracking for non-technical users. This duality, justified by extensive user testing, ensures that the solution remains accessible to all support levels while still providing the advanced relational power required to transform raw interaction data into actionable business intelligence.

2.1.3 Fundamentals of Management and Development Methods

Change management within the project aims to ensure that modifications in scope, schedule, costs, or other aspects are properly controlled and aligned with strategic objectives. In agile methodologies such as Scrum, changes are incorporated iteratively, enabling continuous adaptation based on continuous stakeholder feedback. These changes are managed through the product backlog, where new requirements can be added or existing user stories adjusted at any time. However, once a sprint is initiated, the team seeks to maintain focus on the planned scope, limiting changes during its execution. All change requests are evaluated by the Product Owner, considering their impact on the solution, the value added to the business, and the effort required for development and implementation.

Backlog prioritization is conducted at the end of each sprint with the participation of the Product Owner and relevant stakeholders, defining the priority level of each item and its allocation to future sprints. During Sprint Planning, the team selects the highest-priority backlog items to be developed in the upcoming sprint. Communication management plays a key role in ensuring transparency and alignment among all stakeholders, promoting an efficient flow of information related to project progress, risks, and decisions.

The agile framework is supported by structured ceremonies (rites), which guide execution, monitoring, and continuous improvement throughout the project lifecycle:

A. **Sprint Duration:** Fixed iteration cycle of 2 weeks.
B. **Daily Scrum:** 15-minute daily meeting held on every working day of the sprint, focused on synchronizing the team's activities, identifying impediments, and ensuring alignment with sprint goals.
C. **Sprint Planning:** Conducted on the first day of each sprint, defining the sprint goal, selecting backlog items, and estimating tasks and responsibilities.
D. **Sprint Review**: Held on the last day of the sprint, involving stakeholders to validate deliveries, gather feedback, and identify potential adjustments or new requirements.
E. **Sprint Retrospective:** Performed immediately after the Sprint Review, allowing the team to reflect on what went well, what can be improved, and define action items for the next sprint.

2.2 Specification and Development

2.2.1 Requirements and Specifications

To ensure that the development of the project is perfectly aligned with the strategic needs of BTG Empresas, a clear framework of Requirements and Specifications has been established. This section outlines the specific behaviors the system must exhibit to solve data fragmentation and the operational standards it must meet to ensure stability and security within the bank's ecosystem

2.2.1.1 Functional Requirements

I. FR1 - To achieve these objectives, the project needs to ensure that customer action events are correlated in a structured way within a graph model, allowing efficient navigation between data and business relationships. This correlation will enable unified querying of customer interactions with the bank, reducing information fragmentation and making analysis more accessible and dynamic.

II. FR2 - The interactive dashboard will be one of the main deliverables, providing an intuitive interface that allows the support team to search for SME data via their CNPJ or CPF, enabling them to quickly identify the origin of issues and provide faster solutions. Additionally, it will enable the business team to track behavior trends through intuitive graphs and a panel for building filters and queries, optimizing strategies for personalizing products and services.

III. RF3 - A topic will also need to be created via AWS's SNS (Simple Notification Service), where teams will publish event and user information, and a Lambda function will be responsible for processing and inserting data into the database.

IV. RF4 - Data security will be a critical factor, with strict access management and the definition of specific user profiles-administrator, editor, and viewer-to ensure appropriate permission levels. Only BTG Empresas employees who have registered CPF and password and have received an access invitation via institutional email will be able to view or manipulate the data, ensuring control and protection of the information.

2.2.1.2 Non-Functional Requirements

I. NFR 1 - The solution needs to be scalable enough to handle large volumes of data, especially during periods of multiple user access. To achieve this, optimization and indexing strategies for queries will be implemented, ensuring efficient system response performance.

II. NFR 2 - Additionally, the platform must have high availability, preventing disruptions that could impact customer support and strategic analysis. Robust security measures will be applied, such as secure authentication and access restricted to IPs within BTG Empresas internet network, ensuring that sensitive information is not accessed improperly.

III. NF3 - Finally, the system will be developed with a modular architecture, and the database will follow an object-relational architecture, allowing for future evolutions and improvements without compromising the stability of the solution. This will ensure that the project can be expanded as new analysis and monitoring demands arise.

This initiative aims not only to optimize customer support but also to strengthen data-driven decision-making, ensuring greater scalability, efficiency in internal processes, a reduction in costs, and an increase in customer satisfaction.

2.2.2 Architecture and Technology

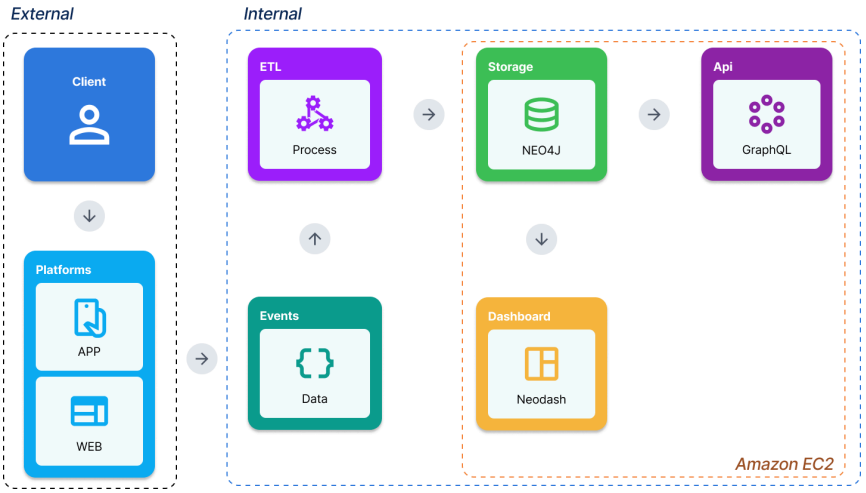
Node Insight utilizes a Cloud-based Client-Server architecture designed for high scalability and modularity. The system is hosted within the Amazon EC2 environment, which provides a secure and resilient boundary for the core processing and storage components.

The architecture is organized into three primary layers as illustrated in the diagram:

I. **Presentation Layer (External):** Users access the system through the BTG Empresas Web and Mobile App platforms. These platforms act as the entry point for all client interaction data.
II. **Data Processing and Storage Layer (Internal):** Incoming interaction data is captured as Events and processed through an ETL (Extract, Transform, Load) layer. This stage normalizes the data for ingestion into Neo4j, a native graph database that stores information as nodes and edges to map complex financial relationships.
III. **Application and Analytics Layer (Internal):** A GraphQL API serves as the flexible query interface, allowing the system to serve specific data needs efficiently. This is complemented by Neodash, which provides the interactive Dashboard for real-time relational visualization.

Below is the figure of the more complete architecture diagram:

Figure 1: Architecture Diagram.



Source: The author (2025).

2.2.2.1 Integration into the IT Ecosystem

The Node Insight solution is designed to integrate seamlessly into BTG Pactual's existing infrastructure to minimize operational friction.

- **Infrastructure Consistency:** The system leverages the bank's existing partnerships with AWS and Azure, ensuring that hosting and tool usage (such as the testing environment and cloud resources) remain consistent with corporate standards.
- **Data Unification:** It integrates directly with the BTG Empresas Portal and App to ingest decentralized data events. By correlating this data in a single graph-oriented database, it reduces information fragmentation across different team silos.

2.2.3 Development and Implementation (MVP)

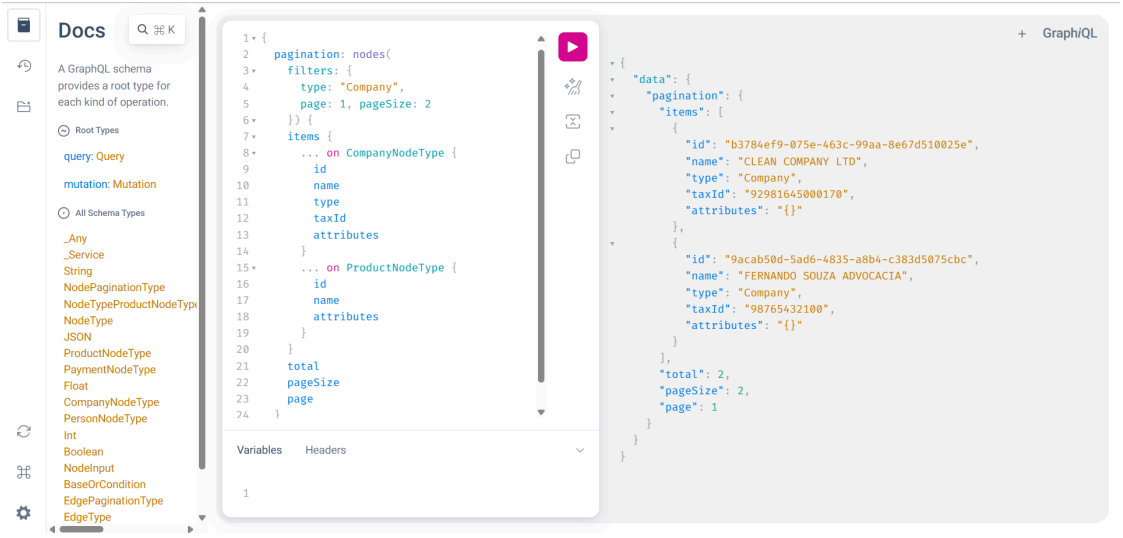
The development of the Node Insight MVP was orchestrated through an Agile framework that harmoniously combined Scrum and Kanban methodologies to ensure iterative delivery and efficient workflow management. By organizing development into two-week sprints, the team maintained a consistent rhythm of delivery supported by standard ceremonies—including Sprint Planning, Daily Scrums, Sprint Reviews, and

Retrospectives—which facilitated continuous stakeholder validation and process improvement. This agile cycle was complemented by a Kanban board managed via GitHub Projects, providing real-time visibility into the backlog and task progression across all development stages. To safeguard code quality, the workflow utilized a Continuous Integration (CI) pipeline powered by GitHub Actions; this automated system triggered static code analysis via Flake8, type checking with Mypy, and unit tests through Pytest upon every commit, ensuring high stability and preventing regressions throughout the project lifecycle.

In terms of infrastructure, the solution was deployed using a robust cloud-native approach hosted within the Amazon EC2 environment, ensuring high availability and seamless integration into the bank's existing IT ecosystem. The system's core storage, a Neo4j graph database, was implemented utilizing AWS Kubernetes pods to allow for horizontal scaling and operational resiliency. Consistency across development, testing, and production machines was achieved through containerization with Docker and Docker Compose, which managed the execution of both the Neo4j backend and the UI application. Furthermore, a serverless ingestion flow was established using AWS SNS and Lambda functions to process and insert real-time interaction data from the BTG Empresas Portal and App directly into the graph database, enabling the transformation of raw events into relational insights.

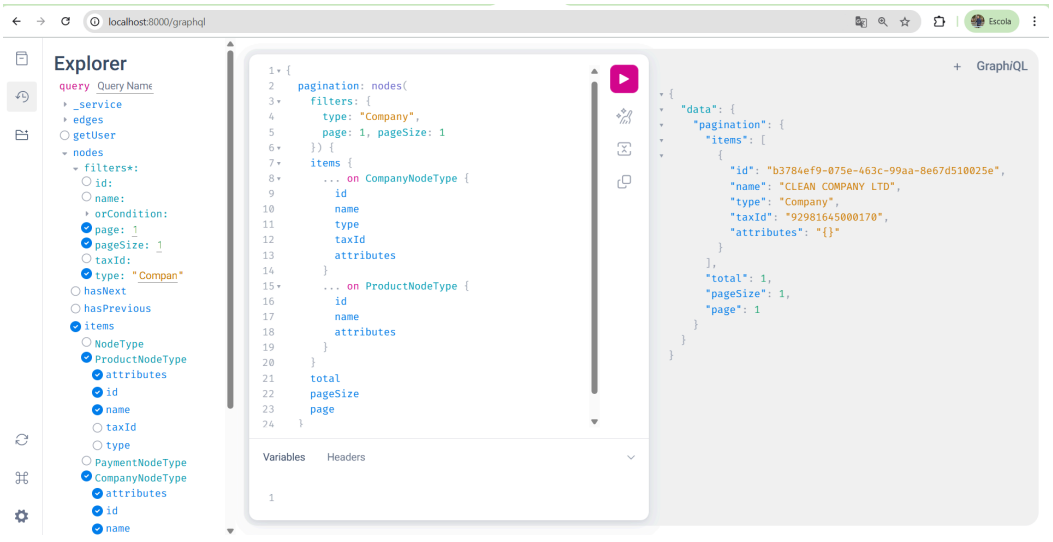
Validation of the initial MVP was conducted using a representative dataset of 20,000 clients - Production (PROD) environment and 5 clients - User Acceptance Testing (UAT), with the database undergoing rigorous testing. This strategic deployment was specifically intended to enable product cross-selling; by mapping intricate relationships between companies and product usage such as Pix, Credit Cards, and Vehicle Debt the solution allows the business team to identify behavior patterns and unmet customer needs with high precision. Finally, the GraphQL API served as the central query interface, implemented using Clean Architecture to decouple core business logic from external drivers and frameworks. This API has already reached an operational milestone, being utilized by several internal teams to query registered node and edge data, allowing them to retrieve tailored insights through a flexible and stable contractual interface.

Figure 2: Server running with API response.



Source: The author (2025).

Figure 3: Example of pagination usage.



Source: The author (2025).

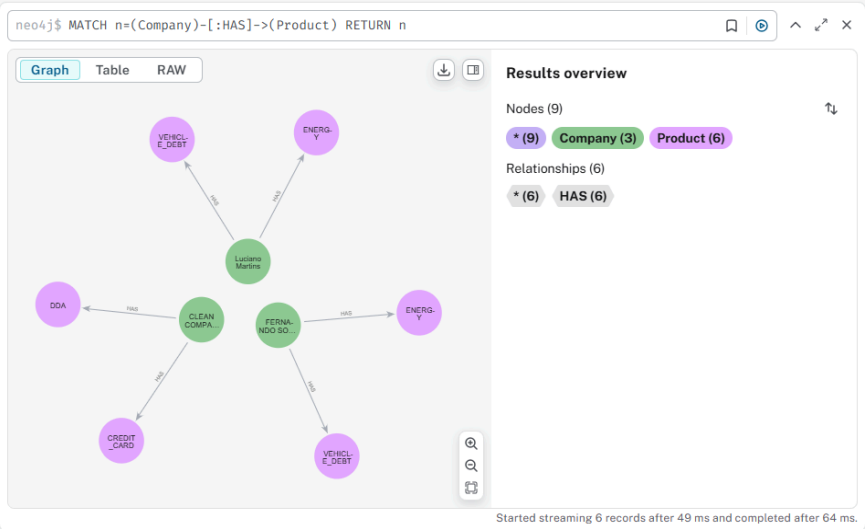
2.2.4 Testing and Technical Evaluation

The testing and technical evaluation phase for the project was designed to ensure that the solution not only met the rigorous technical standards of BTG Empresas but also addressed the human and operational complexities of their support ecosystem. By combining automated code validation, performance benchmarking, and direct user feedback, the project established a comprehensive quality assurance framework.

2.2.4.1 Data Validation and Mock Testing

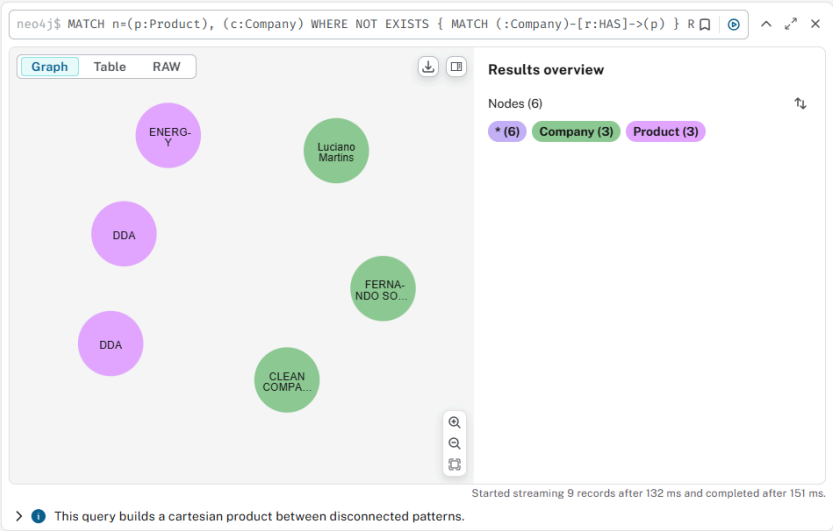
A series of functional tests using accurate data mocks to assess the quality and reliability of the graph structure. Utilizing the Cypher query language, the system successfully demonstrated its ability to handle complex relational requests, such as listing all active products belonging to clients, filtering products by CNPJ/CPF, and generating descending payment lists. These tests confirmed that the node and edge definitions correctly reflect the bank's business rules, allowing for precise data retrieval even when filtering by specific product types or transaction statuses. Below, you can find the results of some tests.

Figure 4: List of all active products belonging to all clients.



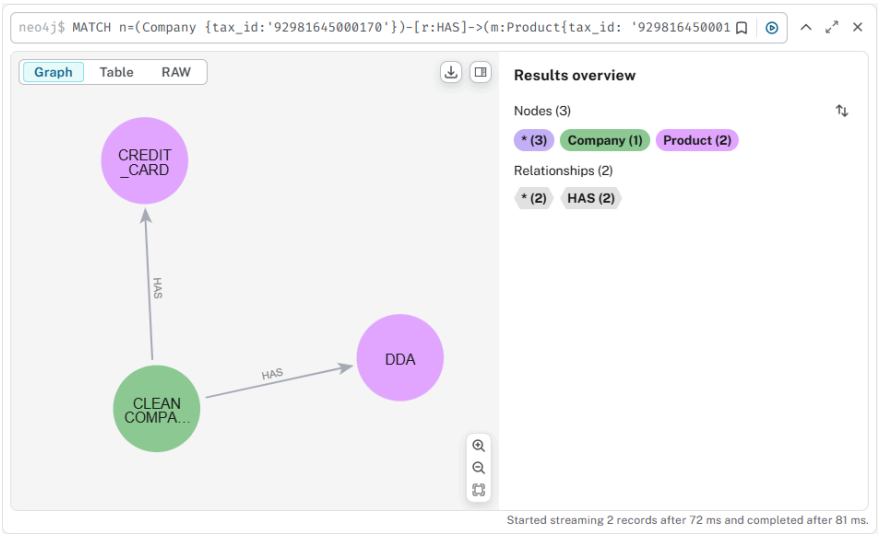
Source: The author (2025).

Figure 5: List of all deactivated products belonging to all clients.



Source: The author (2025).

Figure 6: A list of all active products of a client, filtered by CNPJ/CPF.



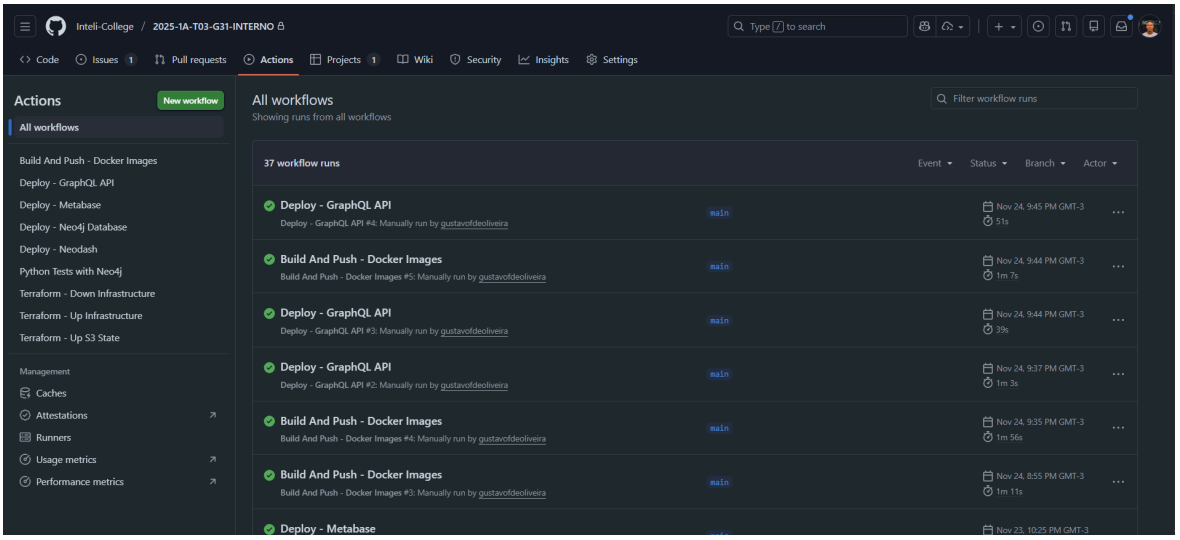
Source: The author (2025).

2.2.4.2 Code Quality and CI/CD Integration

To maintain a high standard of maintainability, a robust suite of unit tests was developed using the Pytest framework. Following the Arrange–Act–Assert paradigm, these tests provided full coverage across the Domain, Use Case, and Assembler layers. Key validation areas included attribute constraints for Node and Edge entities, business rule enforcement for user authentication, and data normalization within the GraphQL assemblers. This entire suite was integrated into a GitHub Actions CI

pipeline. On every commit, the pipeline automatically runs static code analysis via Flake8, type checking with Mypy, and coverage tracking with pytest-cov, ensuring that no regressions reach the production environment.

Figure 7: All CI/CD pipelines developed.



Source: The author (2025).

2.2.4.3 Performance Benchmarking (Grafana K6)

The technical evaluation included a rigorous performance analysis using k6 to compare the resilience of the Node and Edge routes. The results provided a clear picture of the system's current limits:

Test Type	Node Route Results	Edge Route Results
Breakpoint	Saturation at early concurrency; 97% failure rate.	Slightly faster start but 98.5% failure rate under high load.
Load	Stable with 0% failure; high latency (avg 2s).	Stable with 0% failure; slower than Node (avg 5s).

Spike	Survived bursts with 0% failure; severe peaks (up to 52s).	Survived bursts with 0% failure; peak latency up to 54s.
Stress	Collapsed under sustained extreme load; 97.6% failure.	Collapsed under sustained extreme load; 98.9% failure.

The final conclusion of the performance evaluation is that while the routes are reliable for light to moderate usage, optimization of GraphQL resolvers and underlying infrastructure is necessary to handle production-level spikes and higher concurrency.

2.2.4.4 User Acceptance and Support Integration

A critical component of the technical evaluation was the human-centric testing conducted with the BTG support team. Interviews mapped the existing support hierarchy from L1 (initial scripts) to L3 (core development), highlighting the pressure of SLA deadlines and the diversity of technical backgrounds among staff. Practical usability tests with two support analysts provided the following insights:

- **Support 1 (Technical Profile):** Successfully built independent queries but noted that native Neo4j error messages were unclear, making debugging frustrating.
- **Support 2 (Non-Technical Profile):** Highlighted the steep learning curve of Cypher but praised **NeoDash** for its one-click chart refreshes and PNG export capabilities.
- **Shared Feedback:** Both users reported that dense graph visualizations were difficult to interpret and suggested improved color-coding and more intuitive zoom functionality.

These evaluations reinforced the project's goal of balancing technical depth with accessibility. The insights gained are guiding current iterations to simplify visual density in NeoDash while maintaining the powerful relational exploration capabilities that Neo4j provides for advanced users.

2.3 Assessment of Impact and Contribution to the Business

The implementation of the project represents a high-return investment for BTG Empresas, with a projected profit of R$ 3,798,808.98 against a total investment of R$ 451,191.02. By applying a cost-reduction strategy including the adoption of ARM-based architecture and reserved AWS instances, infrastructure expenses are expected to drop by 25% to 30%, further increasing the final projected profit to R$ 3,978,806.78. Beyond direct financial gains, the solution mitigates the risk of SLA breaches, which historically occurred due to manual searches across decentralized databases.

2.3.1 Defining Corporate Success Metrics

To measure the success of Node Insight, specific Key Performance Indicators (KPIs) have been established based on the project's strategic objectives:

- **SLA Breach Reduction Rate:** Measures the decrease in overdue support tickets by providing a unified graph view that eliminates the need for manual cross-database queries.
- **Mean Time to Diagnose (MTTD):** Tracks the average time a support agent (L1/L2) takes to identify the origin of a client issue, such as a failed Pix transaction, using the NeoDash interactive interface.
- **Infrastructure Efficiency Ratio:** Monitors the percentage of cost savings achieved through the transition to ARM architectures and auto-scaling solutions during high-volume periods (e.g., payroll processing).
- **Business Intelligence Adoption:** Measures the frequency of use of the Metabase and NeoDash dashboards by non-technical teams to track product life cycles and customer behavior patterns.
- **Data Standard Compliance:** The percentage of new product data published by developers that strictly adheres to the defined graph model standard.

2.3.1.1 Measurement Methodology

The methodology for evaluating the impact involves a comparative analysis of the "Before" and "After" states of the **BTG Empresas** support and business workflows:

I. **Baseline Data Collection (Before):** Historical data was gathered through interviews with the support team to map existing routines, highlighting the inefficiency of manually consulting multiple databases for a single diagnostic.
II. **Performance Benchmarking:** Technical "After" metrics were collected using Grafana K6 to simulate real-world stress scenarios, such as traffic spikes during FGTS and IPVA processing, to validate system resilience compared to previous unscaled environments.
III. **User Acceptance Testing (UAT):** Practical sessions with support analysts evaluated the usability of the Cypher queries and NeoDash dashboards, providing qualitative data on how the tools reduce the learning curve for non-technical users.
IV. **Financial Validation:** Final profit and cost reduction figures were calculated using a Pareto principle analysis, categorizing returns between small-to-medium businesses and large enterprises to ensure long-term fiscal stability.

2.3.2 Results and Impact Analysis

The implementation of the solution marks a definitive shift from the baseline metrics of data fragmentation and high operational overhead toward a model of high financial and technical efficiency. Quantitatively, the project surpassed the initial viability expectations by achieving a projected profit of R$ 3,978,806.78. This result was primarily driven by a strategic reduction in infrastructure expenditures, which decreased by 25% to 30% compared to the initial projections through the adoption of reserved instances and ARM architectures within the AWS environment. By optimizing the total project investment from an initial R$ 451,191.02 to a leaner R$ 271,193.23, the solution demonstrated a clear mastery over resource allocation and successfully mitigated risks associated with currency fluctuations.

Beyond these direct savings, the qualitative analysis reveals a transformative gain in operational agility for internal support teams, specifically across the L1, L2, and L3 levels. By replacing manual data correlation with a unified 360-degree graph view, the solution improved decision-making and drastically reduced the time required to diagnose complex customer issues, thereby mitigating SLA breaches.

Internal customer satisfaction increased as technical barriers were lowered through the abstraction of complex queries into intuitive dashboards. This allows business teams to shift their focus from technical troubleshooting toward high-value strategic initiatives, such as product cross-selling and behavioral pattern analysis, ultimately strengthening the institution's capacity for data-driven growth.

2.3.3 Cost-Benefit Analysis

2.3.3.1 Financial Analysis

As with every project developed at INTELI, the financial analysis plays a key role in determining whether the initiative is truly viable and makes sense from a business perspective.

Therefore, this analysis was built taking into account the company's specific context and its operational dynamics. Below, you can find the main results and insights derived from this financial assessment.

Service	Note	Allocated People/Services	Cost	Price (CLT/Tax)	Months	Total Price
Senior Analyst		1		R$11,000.00	6	R$66,000.00
Assistants		2		R$10,000.00	6	R$60,000.00
Director		1		R$20,000.00	6	R$120,000.00
Product Owner		1		R$15,000.00	6	R$90,000.00
Neo4j Cloud	Consider 3 months of development + 3 months of testing in the **sa-east-1 (São Paulo)** region.	1	$2,336.00	$3,568.50	6	R$115,191.18
Amazon EKS		1	$153.00		6	
Amazon ECS		2	$184.50		6	

Confluent Kafka		1	$895.00		6	
					Tax	R$8,253.50
					Total with Tax:	R$459,444.68
					Dollar Rate:	R$5.38
					Inclusive Tax Rate:	0.16
					Total with Inclusive Tax:	R$451,191.02

The financial analysis shows that the project is economically viable and offers a significant return for the company. The total costs, considering labor of R$336,000.00 and services of R$115,191.18, combined with an inclusive tax of 16%, amount to R$451,191.02. Based on this, the estimated profit reaches R$3,798,808.98, demonstrating that the required investment is more than covered by the expected returns.

Beyond the numbers, the analysis highlights the strategic relevance of the project, showing that it not only meets the company's operational needs but also provides sufficient margin for future adjustments and expansions.

For full transparency and detailed breakdowns, all calculations, cost distributions, and projections are available in the complete financial analysis Excel file. This document allows stakeholders to review each cost item and simulate different investment and profit scenarios.

2.3.3.2 Infrastructure Cost Reduction Strategy

The cost optimization analysis indicates that the project can achieve significant savings while maintaining operational efficiency. By reallocating team members to other initiatives, the effective project dedication drops to 55% of their time, resulting in a 45% reduction in labor costs. Additionally, through the company's partnership with AWS, further savings are achieved by using reserved instances for

2–3 years and adopting ARM-based architecture, leading to an estimated 25%–30% reduction in cloud service expenses.

These combined measures result in a substantial overall cost reduction, increasing the project's profitability and providing greater flexibility for reinvestment in innovation or expansion. For full transparency, all detailed calculations, cost variations, and reduction scenarios are available in the complete financial analysis file, allowing stakeholders to review and simulate different cost and savings projections.

A complementary exchange rate variation analysis was also conducted to assess the project's financial resilience under different dollar rates. Using a base rate of R$5.38, and simulating a range between R$5.30 (minimum) and R$5.42 (maximum), the total cost with inclusive tax fluctuates between R$269,908.57 and R$271,835.56, compared to R$271,193.23 at the base rate. These minor variations demonstrate that the project remains financially stable even under moderate currency fluctuations, confirming the robustness of the cost optimization strategy.

Finally, the cost reduction strategy significantly enhances the project's profitability. By optimizing labor allocation and leveraging AWS cost efficiencies, the total cost decreases from R$451,191.02 to R$271,193.23, representing a substantial improvement in operational efficiency. Consequently, the projected profit rises from R$3,798,808.98 to R$3,978,806.78.

Based on the Pareto principle, where 80% (R$250,000.00) of clients are small and medium-sized businesses and 20% (R$4,000,000.00) are large enterprises, this increase demonstrates a more balanced and sustainable profit distribution. The results confirm that the cost optimization measures not only reduce expenses but also strengthen long-term financial resilience and return potential.

2.3.4 Critical Success Factors and Lessons Learned

The development of the project provided deep insights into the intersection of graph technology and corporate banking operations. The following factors were instrumental to the project's success, while the challenges encountered offered valuable technical and strategic lessons.

2.3.4.1 Critical Success Factors

A. **Robust Technical Stack and Code Quality:** The choice of Python as the primary language for the GraphQL API was a strategic success, as its mature ecosystem allowed for the implementation of strict quality gates, including static analysis with Flake8 and type checking with Mypy. This ensured that the large and complex data volume was handled by a maintainable and testable codebase.

B. **Native Graph Performance:** Implementing Neo4j was vital for achieving the "360° view" of the customer. Unlike relational alternatives, Neo4j's native graph engine enabled high-performance exploration of interconnected data, such as correlating multiple interaction events to a single company entity without the overhead of complex JOIN operations.

C. **User-Centric Methodology:** Following the Inteli philosophy of "designing for the user, not just for the problem," the project prioritized direct engagement with the BTG Support Teams (L1, L2, and L3). This approach ensured that the final dashboards weren't just technically functional but were tailored to the real-world pressure of SLA deadlines and the diverse technical backgrounds of the staff.

2.3.4.2 Lessons Learned

A. **The Integration Paradox:** A significant lesson was learned during the attempt to connect Metabase directly to Neo4j. The failure of the community-maintained APOC plugin which proved incompatible with modern versions forced a pivot toward a hybrid ecosystem using NeoDash for graph-specific depth and a separate on-premise PostgreSQL for traditional metrics. This highlighted the importance of early compatibility spikes in architectural planning.

B. **The Learning Curve of Cypher**: User testing revealed that while graph data is powerful, the Cypher query language presents a barrier for non-technical users. This led to the critical realization that "raw" database access must be abstracted through intuitive visual interfaces to prevent user frustration and ensure widespread adoption.

C. **Performance Limits under Stress:** Testing with Grafana K6 revealed that both the Node and Edge routes require further optimization in their GraphQL resolvers to handle extreme concurrency and real-world production spikes.

This lesson emphasized that stability under moderate load does not guarantee resilience under the extreme pressure of high-volume processing dates, such as payroll or tax windows.

3 Conclusion

The conclusion of the Node Insight project confirms that the objectives set out during the planning phase were fully achieved through the implementation of a graph-oriented solution that successfully correlates fragmented customer interaction data. By consolidating information from the BTG Empresas Portal and App into a unified Neo4j structure, the project established an integrated view of customer interactions, significantly improving operational efficiency and addressing the critical issue of SLA breaches in the support departments.

Regarding the business impact, the decision to utilize graph data structures brought immediate operational intelligence and cost reduction to the project. As initially hypothesized, this relational foundation was a strategic necessity; attempting to build a dedicated AI team or implement advanced models without such a structured data base could have resulted in expressive spending without achieving positive results. Instead, the current architecture justifies its investment with a projected profit of R$ 3,978,806.78, proving that technical efficiency and financial resilience can be achieved simultaneously.

Furthermore, the solution possesses a strong and significant trend for evolution over time, with the inherent capacity to encompass a wide range of existing and future projects and use cases within the company. Because Node Insight centralizes and standardizes data from various clients, it provides a robust and fertile base for the future implementation of Artificial Intelligence (AI), specifically for advanced patterns such as fraud analysis and churn monitoring. To support this continuous evolution, a full knowledge transfer plan has been established, supported by comprehensive technical documentation available via GraphQL Playground and the project's GitHub repository, ensuring that the internal team can perform maintenance and safely extend the system's capabilities.

References

Book:

SILVA, José Pereira da**. Análise Financeira das Empresas**. 12. ed. São Paulo: Atlas, 2017.

FOWLER, Martin. **Patterns of Enterprise Application Architecture.** Boston: Addison-Wesley, 2002.

MARTIN, Robert C. **Clean Architecture: A Craftsman's Guide to Software Structure and Design.** 1st ed. Boston: Prentice Hall, 2017.

SCHWABER, Ken; SUTHERLAND, Jeff**. The Scrum Guide: The Definitive Guide to Scrum - The Rules of the Game. [S.l.]**: Scrum.org, 2020.

Sites:

AMAZON WEB SERVICES. **AWS Pricing Calculator**. Available at: https://calculator.aws. Accessed on: Dec. 07, 2025.

GLASSDOOR. **Glassdoor: salaries, reviews, and company insights.** Available at: https://www.glassdoor.com. Accessed on: Dec. 07, 2025.

AMAZON WEB SERVICES. **Amazon EC2: User Guide**. 2025. Available at: https://docs.aws.amazon.com/ec2/. Accessed on: Dec 12, 2025.

AMAZON WEB SERVICES. **AWS Lambda: Developer Guide**. 2025. Available at: https://docs.aws.amazon.com/lambda/. Accessed on: Dec 12, 2025.

DRIZZLE TEAM. **Drizzle ORM: Documentation**. 2025. Available at: https://orm.drizzle.team/. Accessed on: Feb 24, 2025.

GRAFANA LABS. **k6 Documentation**. 2025. Available at: https://k6.io/docs/. Accessed on: Dec 09, 2025.

GRAPHQL FOUNDATION. **GraphQL Specification**. Oct 2021 ed. 2025. Available at: https://spec.graphql.org/. Accessed on: April 10, 2025.

METABASE. **Metabase Open Source Documentation**. 2025. Available at: https://www.metabase.com/docs/latest/. Accessed on: Nov 15, 2025.

NEO4J. **Cypher Query Language Manual v5**. 2025. Available at: https://neo4j.com/docs/cypher-manual/current/. Accessed on: Mar 09, 2025.

NEO4J LABS. **NeoDash: Dashboard Builder for Neo4j**. 2025. Available at: https://neo4j.com/labs/neodash/. Accessed on: Mar 09, 2025.

PYTEST DEVELOPMENT TEAM. **Pytest: help you write better programs**. 2025. Available at: https://docs.pytest.org/. Accessed on: May 15, 2025.

Annexes

ANNEX A – Consolidated Financial Analysis

This annex presents the raw economic viability data and infrastructure costs for the project, supporting the investment in human capital and cloud services.

Source: The author (2025).

Link: The complete dynamic spreadsheet with all calculation formulas and detailed cost distributions is available at: Financial Analysis.xlsx.

ANNEX B – Infrastructure Cost Reduction Strategy Report

Supporting documentation for financial efficiency metrics and AWS resource optimization applied to the solution's ecosystem.

Source: The author (2025).

Link: The detailed cost reduction scenarios and exchange rate variation analysis can be reviewed at: Cost Reduction Strategy.xlsx.