

# Database Application - Relatório

Este documento apresenta o desenvolvimento e os resultados do projeto de aplicação de banco de dados conduzido pela equipe DATA FORCE. Inclui uma visão geral do projeto, detalhes sobre a escolha da tecnologia NoSQL, o conjunto de dados utilizado, as etapas de desenvolvimento, os artefatos produzidos e os próximos passos.

DATA FORCE - INTEGRANTES			
Nome	Matrícula	Curso	Função
Anna Júlia	01631263	Sistemas da Computação	Limpeza do Banco de Dados   Dicionário
Kawã Bazante	01616235	Engenharia da Computação	Documentação  Grafo
Nicole Galvão	01602503	Ciências da Computação	Modelo Lógico
Eduardo Oliveira	01618738	Engenharia da Computação	Modelo Relacional
José Paulo	01622179	Engenharia da Computação	

# Introdução

---

Nossa equipe, nomeada como **DATA FORCE**, composta por **cinco membros**, embarcou em uma empolgante jornada de criação de uma aplicação de banco de dados. Nosso objetivo foi explorar e aplicar conceitos de modelagem lógica e física, além de engenharia reversa, utilizando uma tecnologia NoSQL para gerenciar dados de forma eficiente.

Para começar, optamos por utilizar a tecnologia **NoSQL Grafo**, pois ela oferece vantagens específicas para o nosso projeto, como permitir representar e analisar facilmente essas conexões, tornando-se eficiente para descobrir padrões e caminhos de conexão, mesmo em grandes conjuntos de dados. Decidimos trabalhar com um conjunto de dados do Kaggle, selecionando **Amazon Prime Userbase Dataset**, que nos proporcionou uma gama rica de informações para explorar e analisar.

Para organizar nosso trabalho, criamos um repositório no GitHub, onde todos os artefatos do projeto foram armazenados e compartilhados entre os membros da equipe. Além disso, registramos nossas decisões e progressos em um formulário disponibilizado pelo professor.

Durante o desenvolvimento do projeto, cada membro da equipe assumiu responsabilidades específicas, garantindo uma distribuição equitativa de tarefas e contribuições. No dia do primeiro status report, apresentamos nossas atividades realizadas na semana inicial, destacando o papel de cada integrante no projeto.

Ao final do projeto, produzimos uma série de artefatos, incluindo scripts contendo DDLs, DMLs e DQLs utilizadas na manipulação dos dados em um banco de dados relacional MySQL. Além disso, criamos um modelo de dados lógico e um dicionário de dados detalhado do conjunto de dados escolhido, demonstrando nosso entendimento e aplicação dos conceitos estudados em sala de aula.

Para compartilhar nosso conhecimento e experiência, preparamos um tutorial detalhado sobre a tecnologia NoSQL escolhida, abordando os trade-offs de sua utilização e fornecendo um passo a passo para a extração de dados do banco de dados relacional e sua carga em um banco de dados NoSQL.

# Configuração Inicial

---

Estabelecemos o repositório DATA FORCE no GitHub como o epicentro para reunir todos os elementos do nosso projeto. Estruturamos cuidadosamente suas seções para abranger não apenas códigos, mas também documentações e outros recursos cruciais. Todos os membros do grupo foram devidamente designados como colaboradores, o que possibilitou o acesso e a participação ativa de cada um no desenvolvimento do projeto.

Além disso, implementamos uma série de ferramentas indispensáveis, como o MySQL Workbench é uma IDE específica para o desenvolvimento de código. Essas medidas visam assegurar um ambiente de trabalho coeso e propício à produtividade, permitindo que todos os envolvidos tenham as condições ideais para contribuir eficazmente para o sucesso do projeto.

## Desenvolvimento dos Scripts SQL

---

Na etapa de Modelagem do Banco de Dados Relacional, dedicamos tempo e esforço para construir um modelo de dados lógico robusto, utilizando a plataforma MySQL Workbench. Esse processo envolveu uma análise minuciosa do conjunto de dados, identificando entidades, atributos e relacionamentos do conjunto de dados **Amazon Prime Userbase Dataset**.

Adotamos as regras de normalização estudadas em sala de aula para garantir que os dados fossem organizados de maneira eficiente, minimizando redundâncias e evitando anomalias. Além disso, documentamos meticulosamente o modelo de dados em um dicionário detalhado, onde cada entidade e atributo foi descrito minuciosamente, juntamente com suas relações e restrições correspondentes. Esse procedimento visa não apenas garantir a precisão da representação das informações, mas também facilitar a compreensão e a manutenção do banco de dados ao longo do tempo.

## Modelagem do Banco de Dados Relacional

---

O processo de obtenção do script de banco de dados é uma etapa crucial no desenvolvimento de sistemas de informação. Envolve uma série de passos bem definidos e metodológicos, cada um contribuindo para a criação de um banco de dados robusto e eficiente.

A primeira fase desse processo é a análise de requisitos, na qual são identificadas as entidades principais e seus atributos. Essa etapa envolve uma compreensão profunda das necessidades do sistema e dos dados que serão armazenados, garantindo que o modelo de banco de dados seja adequado para suportar as operações desejadas.

Em seguida, vem a modelagem de relacionamentos, na qual são definidas as conexões entre as diferentes entidades do sistema. Esses relacionamentos podem ser de diversos tipos, como um para um, um para muitos ou muitos para muitos, e são fundamentais para garantir a integridade e consistência dos dados.

Após a modelagem de relacionamentos, é realizada a normalização do banco de dados. Esse processo visa reduzir a redundância e eliminar anomalias de atualização, garantindo que o banco de dados esteja organizado de forma eficiente e coerente.

Uma vez normalizado, o próximo passo é a definição de tipos de dados e restrições. Nesta fase, são especificados os tipos de dados que serão utilizados para cada atributo, bem como quaisquer restrições de integridade referencial ou de domínio que sejam necessárias para garantir a consistência dos dados.

Após a definição de tipos de dados e restrições, o script de banco de dados passa por um processo de refinamento, no qual são feitas otimizações para melhorar o desempenho do sistema. Isso pode envolver a criação de índices, particionamento de tabelas ou outras técnicas para aumentar a eficiência das consultas e operações de manipulação de dados.

Uma vez refinado, o script de banco de dados é testado com dados de amostra para validação. Isso garante que o banco de dados esteja funcionando conforme o esperado e que não haja erros ou problemas de integridade nos dados.

## Modelagem do Banco de Dados Lógico

---

Ao elaborar o script do modelo lógico, foi utilizado recursos de aprendizado online, como vídeo aulas, a fim de facilitar a compreensão e agilizar o processo de produção. Durante esse processo, focou-se em identificar as chaves primárias e estabelecer as relações necessárias entre as entidades. Utilizou-se também a plataforma BR Modelo Web para a construção do modelo lógico, aproveitando suas funcionalidades para iterar e ajustar o design até alcançar o resultado desejado.

Através das vídeo aulas, foi possível absorver conceitos-chave e boas práticas na modelagem de bancos de dados, o que permitiu tomar decisões mais fundamentadas durante o desenvolvimento do script do modelo lógico. As chaves primárias de cada entidade foram anotadas cuidadosamente, garantindo a integridade e eficiência do modelo. Além disso, foram estabelecidas as conexões necessárias entre as entidades,

construindo relações claras e coerentes para representar adequadamente a estrutura e o fluxo de dados do sistema.

Com o auxílio da plataforma BR Modelo Web, foi possível visualizar e manipular o modelo lógico de forma interativa, realizando ajustes conforme necessário para otimizar a organização e a eficácia do banco de dados. A capacidade de iterar rapidamente sobre o design foi crucial para refinar o modelo até alcançar o resultado final desejado.

Em resumo, a combinação de recursos de aprendizado online, como vídeo aulas, e o uso da plataforma BR Modelo Web foram fundamentais para a criação de um modelo lógico robusto e bem estruturado, capaz de atender às necessidades do projeto de forma eficiente e precisa.

## Escolha da Tecnologia NoSQL

---

Durante o processo de Escolha da Tecnologia NoSQL, conduzimos uma pesquisa abrangente sobre as diversas opções disponíveis, que incluem Documento, Colunar, Chave e Valor, e Grafo. Avaliamos cuidadosamente os pontos positivos e negativos de cada tecnologia, levando em consideração os requisitos específicos do projeto em questão.

Após uma análise ponderada, concluímos que a tecnologia NoSQL mais apropriada para atender às necessidades do projeto é o **Grafo**. Essa decisão foi baseada em diversos fatores, como sua flexibilidade de esquema e capacidade de lidar eficientemente com dados semiestruturados, entre outros aspectos relevantes que contribuem para a eficácia e sucesso do projeto.

## Extração e Carga dos Dados no Banco de Dados NoSQL

---

Na etapa de Extração e Carga dos Dados no Banco de Dados NoSQL, empregamos os scripts que desenvolvemos anteriormente para extrair os dados do banco de dados relacional MySQL e importá-los para o banco de dados NoSQL, utilizando Python como a ferramenta principal. Durante esse processo, foi necessário transformar os dados para que se adequassem ao modelo de dados NoSQL e executar operações de carga em larga escala.

Após a conclusão da carga dos dados, realizamos uma validação minuciosa para garantir sua integridade. Isso incluiu a comparação dos resultados obtidos com os dados

originais e a realização de testes abrangentes de integridade. Esse procedimento rigoroso é essencial para assegurar a qualidade e a consistência dos dados no novo ambiente NoSQL, proporcionando uma base sólida para as etapas subsequentes do projeto.

## Extração e Carga dos Dados no Banco de Dados NoSQL

---

Durante a etapa crucial de preparação da apresentação final, cada detalhe foi meticulosamente revisado e aprimorado para garantir um resultado impecável. Reunimos todos os artefatos produzidos ao longo do projeto, desde o modelo de dados até os scripts SQL, a documentação detalhada e o tutorial da tecnologia NoSQL escolhida. Cada elemento foi cuidadosamente organizado, não apenas para transmitir informações de forma clara e acessível, mas também para refletir profissionalismo e excelência.

Além disso, dedicamos tempo considerável para elaborar a apresentação final do projeto. Praticamos a exposição do conteúdo repetidamente, refinando a forma como cada conceito seria apresentado para garantir que fosse compreensível e envolvente para o público. A apresentação abrangeu não apenas uma visão geral abrangente do projeto, mas também detalhes específicos sobre a tecnologia NoSQL selecionada, destacando seus benefícios e aplicações no contexto do projeto. Para complementar, incluímos uma demonstração prática da extração e carga dos dados, proporcionando uma experiência interativa e tangível aos espectadores.

Essa abordagem minuciosa e proativa na preparação da apresentação final reflete nosso compromisso com a qualidade e a excelência em todas as fases do projeto. Estamos confiantes de que nossa apresentação não apenas transmitirá de forma eficaz os resultados alcançados, mas também demonstrará o impacto significativo e o valor agregado do trabalho realizado.

### Passo a passo

Primeiro, escrevemos um código em Python para extrair os dados do banco de dados relacional MySQL, conectando-nos ao MySQL e ao Neo4j usando bibliotecas como `mysql.connector` e `py2neo`. Em seguida, recuperamos os dados do MySQL e os transformamos em um formato adequado para inserção no Neo4j, usando consultas Cypher para criar nodos e relacionamentos com base nos dados extraídos do MySQL.

Após a migração dos dados, escrevemos uma consulta Cypher para criar os relacionamentos desejados no Neo4j. Usando a cláusula `MATCH`, encontramos os nodos relevantes no Neo4j (nodos de `Usuario` e `UsoServico`), filtrando apenas os nodos que compartilham o mesmo `usuario_id` com a cláusula `WHERE`. Em seguida, usamos a cláusula `CREATE` para criar relacionamentos entre os nodos encontrados.

Então, executamos a consulta Cypher no Neo4j Browser ou em nossa aplicação, onde o Neo4j processou a consulta e criou os relacionamentos entre os nodos de Usuario e UsoServico que compartilham o mesmo usuario\_id. Após a execução bem-sucedida da consulta, pudemos verificar os resultados diretamente no Neo4j Browser ou através de consultas no Neo4j Desktop.

Finalmente, verificamos os resultados no Neo4j usando consultas Cypher no Neo4j Browser ou no Neo4j Desktop, como `MATCH (u:Usuario)-[:USA]->(s:UsoServico) RETURN u, s`, para verificar se os relacionamentos foram criados corretamente.

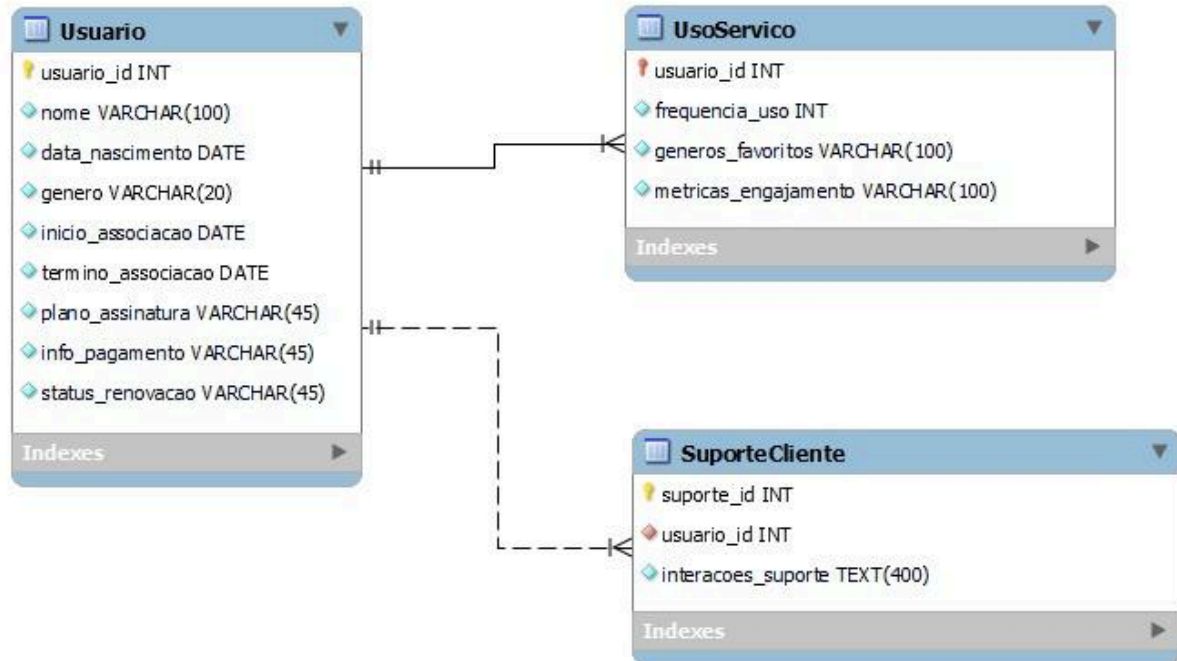
## Conclusão

---

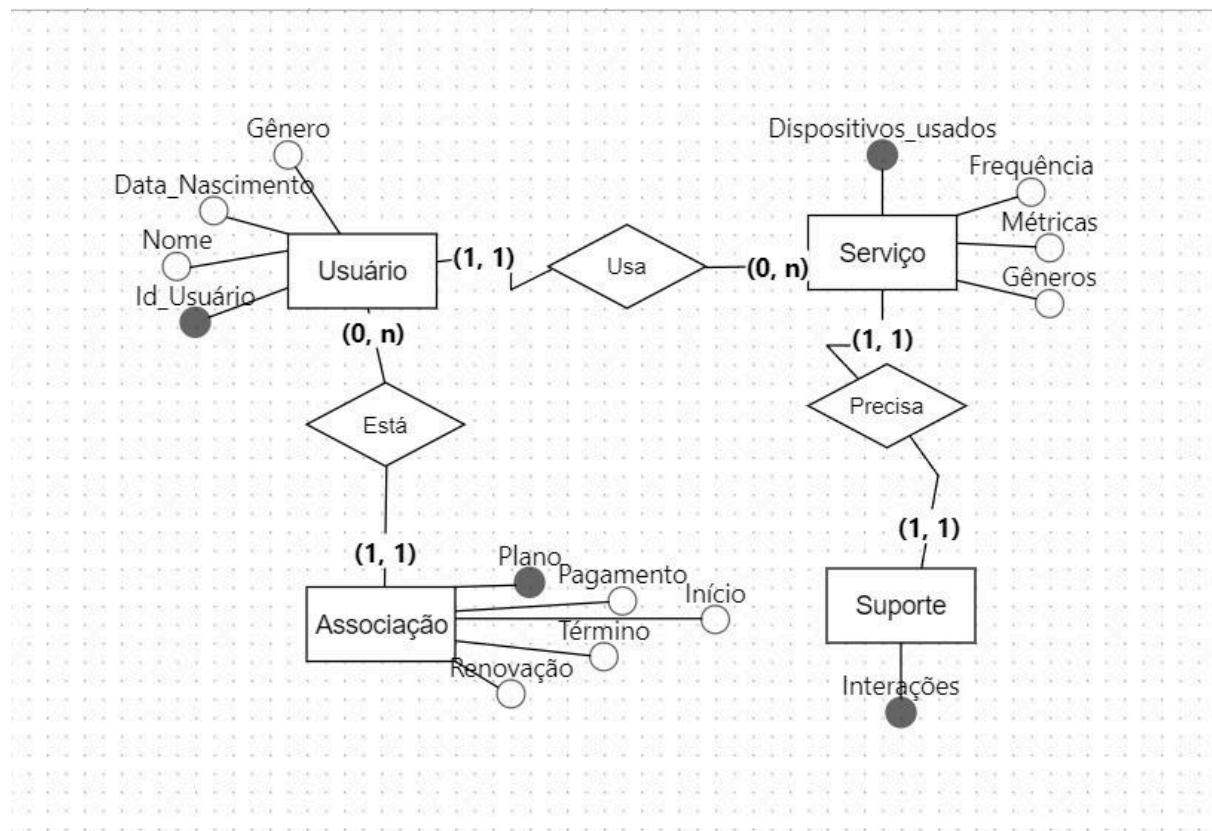
Com grande satisfação, a equipe DATA FORCE concluiu de forma bem-sucedida o projeto de aplicação de banco de dados, garantindo o cumprimento integral de todos os critérios estabelecidos pela atividade de avaliação. Os resultados alcançados refletem nosso compromisso com a excelência e a dedicação de cada membro da equipe. Estamos ansiosos para receber seu valioso feedback e continuar a aprimorar nosso trabalho no futuro.

# Repositório de imagens

## Modelo Relacional

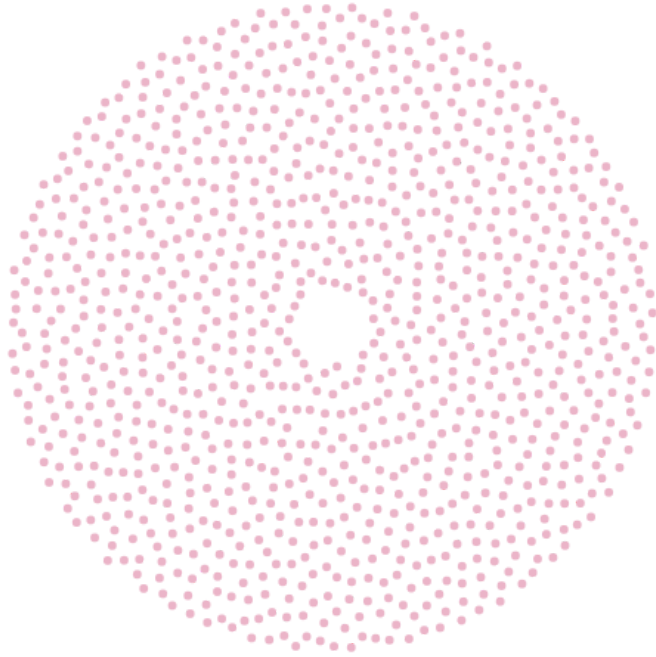


## Modelo lógico





GRAFO, USANDO O: **MATCH (n) RETURN n**



GRAFO, usando: **MATCH (u:Usuario)-[r]->(n) RETURN u, r, n**

