

Reformulação do Ambiente de Dados do Machine Teaching

Renato S. do Nascimento¹, Vitor L. G. C. de Carvalho¹,
Carla A. D. M. Delgado¹, Vivian S. Silva¹

¹Instituto de Computação – Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro, RJ – Brazil

{renatosn, vitorlgcc, carla, vivian}@ic.ufrj.br

Abstract. *Machine Teaching is a platform designed to support the teaching of programming by enabling the automatic grading of exercises. One of the system's key features is a set of dashboards that allow analyses capable of supporting decision-making in the pedagogical context. However, the current dashboard construction process is inefficient, since queries are executed directly on a normalized relational database, resulting in long response times for generating the visualizations. The proposed solution was the development of a dedicated analytical environment through the creation of a data mart based on a dimensional model. This solution, however, encountered another challenge: the lack of documentation for the transactional database, which hindered the mapping of data required for the data mart project. Therefore, a broader approach to redesigning the data environment was proposed, including database documentation, comprising the conceptual and logical models and a data dictionary, in addition to the previously mentioned dedicated analytical environment. This work presents the results obtained so far, encompassing both the process and the products related to the database documentation, as well as the first phase of the data mart development, pointing towards the next stage of the project.*

Resumo. *O Machine Teaching é uma plataforma de apoio ao ensino de programação que permite a correção automática de exercícios. Uma importante funcionalidade do sistema é um conjunto de dashboards, que permitem análises que podem apoiar a tomada de decisão no âmbito pedagógico. Entretanto, a construção dos dashboards atualmente é feita de forma pouco eficiente, dado que as consultas são feitas diretamente a um banco de dados relacional normalizado, resultando em um tempo elevado para a criação das visualizações. A solução proposta foi o desenvolvimento de um ambiente analítico dedicado, com a criação de um data mart baseado em um modelo dimensional. Esta solução, porém, esbarrou em outro problema: a ausência de documentação do banco de dados transacional, que dificultava o mapeamento dos dados para o projeto do data mart. Foi proposta então uma abordagem mais ampla de reformulação do ambiente de dados, com a documentação do banco, incluindo modelos conceitual e lógico, dicionário de dados, além do já mencionado ambiente analítico dedicado. Este trabalho apresenta os resultados já obtidos, englobando o processo e os produtos da documentação do banco, e a primeira fase da construção do data mart, apontando para a próxima etapa do projeto.*

1. Introdução

A aprendizagem de programação é reconhecidamente uma tarefa complexa, envolvendo diferentes processos cognitivos e desafios relacionados à abstração, resolução de problemas e construção de modelos mentais consistentes (Nikula et al., 2011; Rogalski, 1993). Compreender a mobilização cognitiva associada a esse processo é fundamental para aprimorar as situações de ensino-aprendizagem e apoiar estudantes em suas dificuldades iniciais. Nesse contexto, a análise de registros de interação de estudantes

em ambientes digitais tem se consolidado como uma importante fonte de evidências para investigação do processo educacional, permitindo identificar padrões de aprendizagem e comportamento e dificuldades recorrentes.

A plataforma *Machine Teaching* (Moraes et al., 2021) é um ambiente *web* de apoio ao ensino introdutório de Programação, desenvolvido na Universidade Federal do Rio de Janeiro e utilizado desde 2018 em disciplinas introdutórias da área. O sistema foi concebido com o objetivo principal de coletar dados sobre o processo de aprendizagem de programação para apoiar decisões pedagógicas de curto e longo prazo, permitindo identificar pontos de dificuldade e oportunidades de melhoria no processo de ensino-aprendizagem. Além desse caráter estratégico e analítico, a plataforma também possui forte foco operacional, oferecendo funcionalidades para prática de exercícios, correção automática de códigos e fornecimento de *feedback* imediato aos estudantes, o que é um incentivo ao estudante para interagir com a plataforma e, dessa forma, gerar registros do seu processo de aprendizagem.

Estudos recentes na área de *learning analytics* destacam a importância de aproximar os resultados de pesquisa das práticas reais de uso, oferecendo mecanismos capazes de transformar grandes volumes de dados educacionais em informações úteis e acionáveis para docentes, tutores e gestores acadêmicos (Pardo et al., 2018; Rosé et al., 2019; Jovanović et al., 2020). Para que isso seja possível, é necessário não apenas coletar os dados educacionais, mas também organizá-los e disponibilizá-los de maneira adequada para processos de análise e apoio à tomada de decisão. Assim, ambientes educacionais como o *Machine Teaching* passam a demandar, além de infraestrutura operacional para execução das atividades cotidianas, mecanismos especializados para análise histórica e inteligência de dados, capazes de sustentar consultas analíticas mais complexas e apoiar iniciativas de *business intelligence*.

No *Machine Teaching*, estes mecanismos são materializados no formato de *dashboards*, que agrupam uma série de visualizações voltadas tanto para a análise individual de desempenho por parte do aluno, como para o acompanhamento de métricas do curso por parte do docente, seja por aluno ou por turma. Do ponto de vista educacional, estas análises permitem identificar pontos de melhoria, seja em materiais didáticos, na busca ativa de alunos com baixo desempenho ou no processo de condução do curso. Apesar da grande utilidade dos *dashboards*, a versão atual apresenta um problema de eficiência, com tempos elevados para a geração das visualizações. Esta demora, que pode desestimular o uso desta funcionalidade, deve-se ao fato de que atualmente não há ambiente analítico dedicado, e as consultas analíticas, naturalmente mais complexas e intensivas em dados, são feitas diretamente no banco transacional. Este cenário, além de resultar em um alto tempo de resposta para as consultas necessárias na geração dos *dashboards*, pode também acarretar em impactos no ambiente transacional, gerando lentidão nas funcionalidades básicas da plataforma.

A solução proposta inicialmente foi o desenvolvimento de um ambiente analítico dedicado, com a criação de um *data mart* baseado em um modelo dimensional, para melhor desempenho das consultas analíticas e consequente redução de impacto no ambiente transacional. O projeto do ambiente analítico inclui também o desenvolvimento dos processos de ETL (*Extract, Transform, Load* - Extração,

Transformação e Carga) para levar os dados do ambiente transacional para o analítico. Tanto a modelagem do *data mart* quanto o planejamento do ETL exigem a compreensão dos dados disponíveis no ambiente transacional. Porém o *Machine Teaching* não possuía documentação do banco de dados relacional, o que acabou gerando um gargalo na solução proposta. Desta forma, este trabalho apresenta uma abordagem mais ampla de reformulação do ambiente de dados, com a documentação do banco, incluindo a elaboração dos modelos conceitual e lógico correspondentes ao modelo físico implementado, o dicionário de dados, e um diagnóstico de problemas com sugestões de melhorias, além do já mencionado ambiente analítico dedicado, incluindo o projeto e implementação do *data mart* e do processo de ETL.

2. Documentação do Banco de Dados

O banco de dados transacional da aplicação é baseado no SGBD *PostgreSQL* e contém os dados associados às funcionalidades da plataforma, sendo atualmente utilizado tanto para operações transacionais quanto para consultas analíticas. O processo metodológico partiu do modelo físico existente e utilizou diferentes ferramentas para apoiar a engenharia reversa e a documentação do banco, conforme descrito na Figura 1. Inicialmente, esse banco foi analisado em um ambiente containerizado com a ferramenta *Docker*, através de uma instância local. O *DBeaver* foi a ferramenta de gerenciamento de banco de dados utilizada para inspecionar o banco, visualizar tabelas, atributos, chaves, relacionamentos e registros existentes. A partir do recurso nativo de visualização de diagramas desta ferramenta, o modelo lógico do banco foi gerado e posteriormente ajustado manualmente para melhor visualização e agrupamento de domínios relacionados. O esquema analisado continha 52 tabelas, distribuídas entre objetos de infraestrutura do *framework* da aplicação, autenticação e autorização, regras de negócio da plataforma, registros históricos e tabelas depreciadas. Para organizar a análise, as tabelas foram agrupadas em *clusters* conforme seus prefixos, para facilitar a distinção de suas funções nas próximas etapas.



Figura 1: Esteira de execução do processo de engenharia reversa do banco de dados do Machine Teaching.

2.1. Dicionário de Dados

O dicionário de dados foi criado a partir da extração das tabelas dos modelos físico e lógico do banco. Inicialmente, foram gerados arquivos tabulares contendo os nomes dos atributos de cada tabela, tipos de dados e informações de nulidade, em formato CSV, através de um script em *Python* e *Pandas* que extraiu do modelo físico informações referentes ao esquema, aos atributos e às restrições de integridade do banco. Em seguida, esses arquivos foram organizados em planilhas, separadas em abas por *cluster*. Para acelerar a produção das descrições dos campos, foi utilizada uma integração com a API do *OpenAI*, cujo objetivo foi gerar descrições iniciais para os atributos das tabelas,

a partir do modelo físico. Essas descrições foram posteriormente revisadas e complementadas com valores observados no banco e observações adicionais sobre o significado dos campos, utilizando como base tanto o modelo físico como o lógico. A coluna de observações foi continuamente ajustada a partir de alinhamentos com os responsáveis pela criação do modelo físico. Por fim, cada campo de cada tabela ficou descrito com o nome do atributo, tipo, nulidade, descrição, valores encontrados e observações adicionais. Após a geração do modelo conceitual, que será descrito na próxima Seção, o dicionário foi atualizado com a associação dos campos do modelo lógico para o conceitual.

2.2. Modelo Conceitual

A construção do modelo conceitual foi realizada por meio de engenharia reversa a partir do modelo lógico e do dicionário de dados, utilizando a ferramenta *brModelo*. Diferentemente do modelo lógico, que reflete diretamente a estrutura relacional do banco, o modelo conceitual buscou representar os principais elementos do domínio do *Machine Teaching* em termos de entidades, atributos e relacionamentos. A ideia aqui foi repensar a aplicação de forma orientada a dados e sua importância para o negócio. Para isso, foram consideradas majoritariamente as tabelas de regras de negócio. Tabelas auxiliares, históricas, depreciadas ou de infraestrutura foram avaliadas separadamente, fora do escopo da representação conceitual. O processo também envolveu normalização, remoção de redundâncias e balanceamento entre fidelidade ao modelo lógico existente e boas práticas de modelagem.

Como resultado, foi criado um diagrama Entidade-Relacionamento, que pode ser visualizado na Figura 2. Além do modelo conceitual, foi elaborada uma documentação geral da modelagem, reunindo, além do modelo conceitual, o modelo lógico, o dicionários de dados e observações diagnósticas sobre possíveis melhorias no banco.

3. Desenvolvimento do Ambiente Analítico

A documentação do banco de dados, descrita na Seção anterior, permitiu uma ampla compreensão do conjunto de dados disponíveis. Com esta informação em mãos, foi possível dar início ao projeto do ambiente analítico dedicado, descrito a seguir.

3.1. Levantamento de Requisitos de Dados

O levantamento de requisitos de dados foi feito com base nos dashboards já disponibilizados pelo *Machine Teaching*. A partir da análise visual dos 19 painéis, e com o auxílio da documentação e de consultas ao banco transacional para amostragem dos dados, foi possível determinar quais elementos de dados estavam sendo utilizados em cada visualização. Estes dados foram então catalogados, servindo como base para a versão inicial do modelo dimensional. Cada atributo identificado foi classificado como dado quantitativo (que podem ser expressos por meio de métricas numéricas) ou qualitativo (que descrevem e categorizam informações), possibilitando sua tradução em métricas de uma tabela fato ou atributos de uma tabela dimensão, os principais elementos de um modelo dimensional.

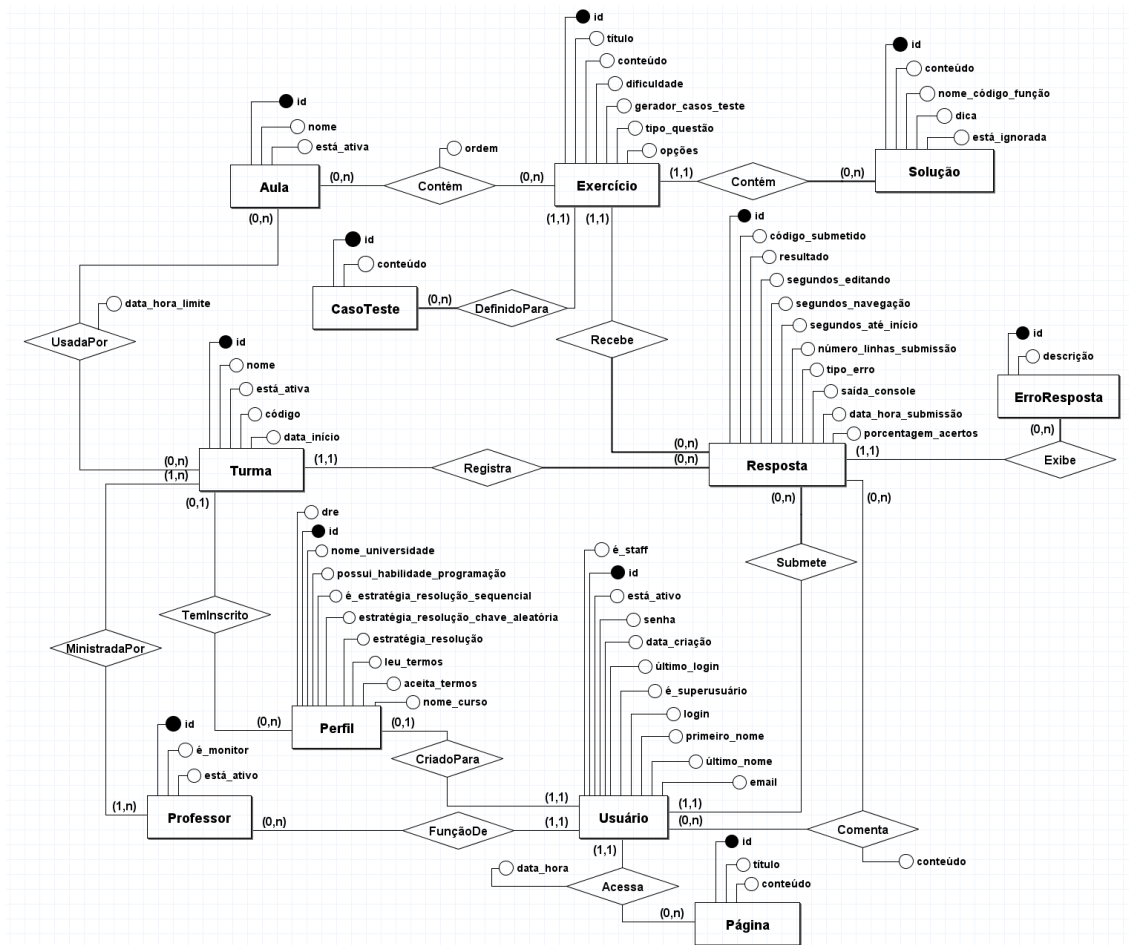


Figura 2: Modelo conceitual resultante do processo de engenharia reversa do Machine Teaching.

3.2. Modelagem Multidimensional

A partir do levantamento de requisitos de dados, foi desenvolvido um modelo contendo, inicialmente, um único fato, considerado suficiente para responder a todas as questões analíticas materializadas nos *dashboards* disponíveis no sistema. A granularidade da tabela fato é uma submissão de resposta a um exercício, ou seja, cada resolução de um exercício enviada por um aluno consiste em um registro nesta tabela. A partir da catalogação dos atributos, foram identificadas nove dimensões que caracterizam o fato: Aluno, Turma, Exercício, Resultado e Erro, além da dimensão Tempo, dividida em data e hora, além de duas dimensões ponte auxiliares. As dimensões *dim_tempo* e *dim_hora_minuto* atuam como dimensões de múltiplos papéis, uma vez que se conectam ao fato mais de uma vez (determinando data e hora do prazo de entrega da resolução, e data e hora da submissão propriamente dita). A única dimensão N:N é a dimensão Erro, dado que uma submissão pode gerar vários erros distintos, por isso a utilização de dimensão ponte. O modelo dimensional é apresentado na Figura 3.

Com a documentação do modelo relacional e a elaboração do dimensional, foi possível dar início ao planejamento do processo de ETL, com o mapeamento entre os elementos de dados dos dois ambientes, bem como o conjunto de transformações, tratamentos e limpezas necessárias para popular o futuro *data mart*, o que inclui tratar eventuais

inconsistências nos dados, realizar junções e/ou uniões de tabelas e padronizar formatos, entre outros procedimentos destinados a garantir a qualidade dos dados e seu melhor aproveitamento nos processos analíticos. O resultado é o plano de alto nível de ETL, que servirá como entrada para a próxima fase do projeto, dividida em duas etapas: carga histórica do *data mart*, para levar todos os dados existentes, e carga incremental em tempo real, para que os *dashboards* sempre reflitam as últimas atividades realizadas pelo alunos na plataforma.

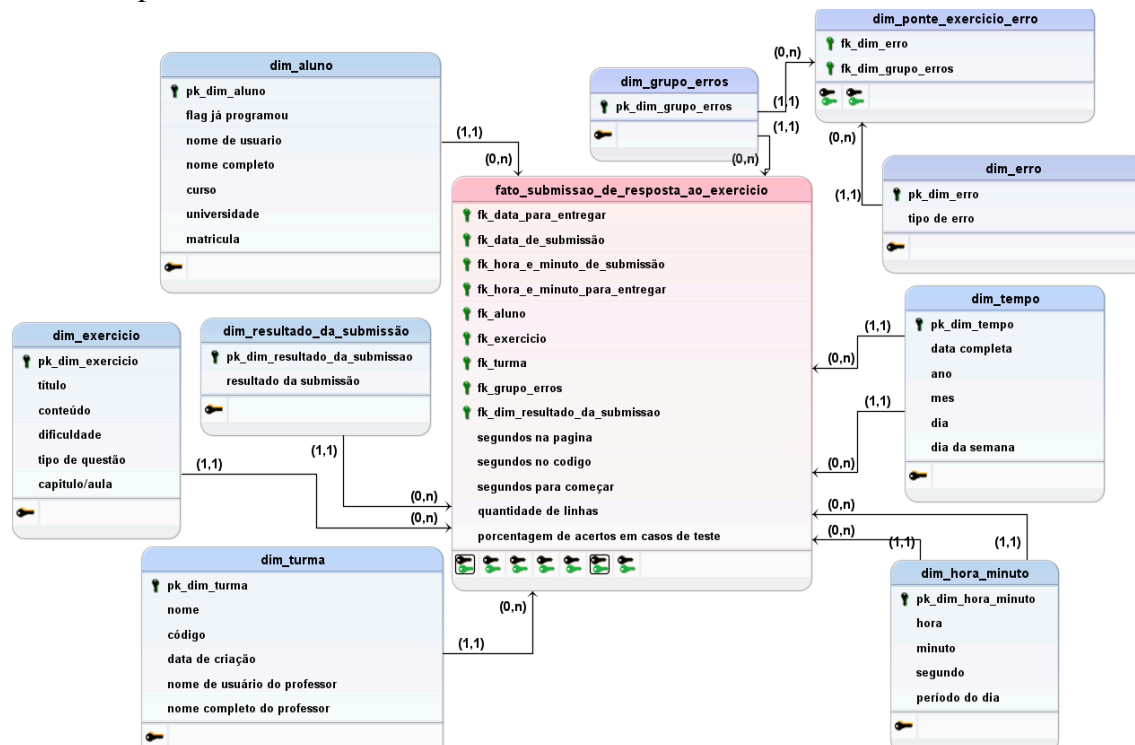


Figura 3: Modelo dimensional para o ambiente analítico do Machine Teaching

4. Conclusão

O *Machine Teaching* é uma importante ferramenta não só de apoio ao ensino de programação, mas também de coleta de dados sobre o processo de aprendizagem para apoiar a tomada de decisões pedagógicas no âmbito deste processo. O conjunto de *dashboards* disponibilizados na plataforma é hoje a principal ferramenta que subsidia a perspectiva de inteligência de negócio da plataforma, porém corre o risco de subutilização devido a problemas de desempenho.

Este trabalho propôs uma reformulação do ambiente de dados do *Machine Teaching*, com a documentação do banco de dados transacional, incluindo modelos lógico e conceitual e dicionário de dados, antes inexistentes, para posterior projeto e implementação de um ambiente analítico dedicado. Enquanto este ambiente dedicado vai permitir a realização de consultas analíticas mais eficientes, melhorando a experiência do usuário no uso dos *dashboards*, a documentação do banco transacional traz vantagens para a plataforma como um todo, facilitando sua manutenção e garantindo padronização nas suas extensões evolutivas.

Declaração de uso de IA. No desenvolvimento deste trabalho foi utilizada uma ferramenta de IA generativa para geração da versão inicial das descrições no dicionário de dados do banco. Nenhuma ferramenta desta natureza foi utilizada na escrita deste artigo.

Origem do trabalho. Este trabalho foi desenvolvido como parte do projeto de Iniciação Científica intitulado “Construção de um ambiente analítico para acompanhamento do desempenho de estudantes”, conduzido no Instituto de Computação da Universidade Federal do Rio de Janeiro.

Referências

- JOVANOVIĆ, Jelena et al. Supporting actionable intelligence: reframing the analysis of observed study strategies. In: *Proceedings of the 10th International Conference on Learning Analytics & Knowledge (LAK '20)*. New York: ACM, 2020. p. 161–170. DOI: 10.1145/3375462.3375474.
- MORAES, Laura O. et al. Supporting Decisions Using Educational Data Analysis. In: *Anais Estendidos do Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia 2021)*. Porto Alegre: Sociedade Brasileira de Computação, 2021. DOI: 10.5753/webmedia_estendido.2021.17622.
- NIKULA, Uolevi; GIEDRAITYTĖ, Vilma; SUKKA, Seppo. Introducing students to programming: a literature review. In: *Proceedings of the Baltic Sea Conference on Computing Education Research*. Koli National Park, Finland: ACM, 2011. p. 9–16.
- PARDO, Abelardo et al. OnTask: Delivering Data-Informed, Personalized Learning Support Actions. *Journal of Learning Analytics*, v. 5, n. 3, p. 235–249, 2018. DOI: 10.18608/jla.2018.53.15.
- ROGALSKI, Janine. Apprenticeship in problem solving and the development of human expertise. In: KAIL, Robert; IZARD, Catherine (org.). *Human Development*. New York: Springer, 1993. p. 91–109.
- ROSÉ, Carolyn P. et al. Explanatory learner models: Why machine learning (alone) is not the answer. *British Journal of Educational Technology*, v. 50, n. 6, p. 2943–2958, 2019. DOI: 10.1111/bjet.12858.