



# Um relato da experiência de construir um sistema para integração de dados e criação de indicadores para um programa governamental

Henrique Varella Ehrenfried\*, Vinicius Tikara Venturi Date\*, Thomas Bianchi Todt\*, Pedro Willian Aguiar\*, Pedro Henrique Kochinski Silva\*, Nicolas André Rizzardi\*, Maria Carolina Sauer\*, Nadja César Ianzer Rodrigues<sup>†</sup> and Eduardo Todt\*

\*Centro de Computação Científica e Software Livre (C3SL)

Universidade Federal do Paraná, Paraná, Curitiba

Email: {hvehrenfried, vtvd19, tbt20, pwa21, phks20, nar20, mcs22, todt}@inf.ufpr.br

<sup>†</sup>Fundo Nacional de Desenvolvimento da Educação

Distrito Federal, Brasília

Email: nadja.rodrigues@fnde.gov.br

**Abstract**—The Brazilian government has several programs to improve the life of Brazilian society, such as the Sistema Único de Saúde (SUS), the Programa Nacional do Livro e do Material Didático (PNLD) and the Programa Universidade Para Todos (PROUNI). These programs have periodic rule revisions, which affect computational systems, causing new software and modules to be created. With new applications, new databases can emerge, a fact that makes it difficult to create indicators about government programs. To this end, a system for integrating data and creating indicators for open source government programs was developed and the experience of this project is reported in this paper. With this experience, new high-impact projects will be able to assess whether or not the related techniques can be used in these new projects.

**Keywords**—Experience report; Development methodology; Indicators panel; Data science.

**Resumo**—O governo brasileiro possui diversos programas para melhorar a vida da sociedade brasileira, como o Sistema Único de Saúde (SUS), o Programa Nacional do Livro e do Material Didático (PNLD) e o Programa Universidade Para Todos (PROUNI). Esses programas tem revisões periódicas de regras, o que afeta sistemas computacionais, fazendo com que novos softwares e módulos sejam criados. Com novas aplicações podem surgir novos bancos de dados, fato que dificulta a criação de indicadores sobre os programas governamentais. Para tanto, foi desenvolvido um sistema de integração de dados e criação de indicadores de programas governamentais de código aberto. A experiência deste projeto é relatada nesse artigo. Com esta experiência, novos projetos de grande impacto poderão avaliar se as técnicas relatadas podem ou não ser empregadas nesses novos projetos.

**Palavras-chave**—Relato de experiência; Metodologia de desenvolvimento; Painel de indicadores; Ciência de dados.

## I. INTRODUÇÃO

O Estado brasileiro tem vários programas que almejam melhorar a vida da sociedade brasileira. Exemplos desses programas são: Sistema Único de Saúde (SUS), Programa Nacional do Livro e do Material Didático (PNLD) e o Programa Universidade Para Todos (PROUNI).

Programas de grande impacto, como os supracitados, costumam ter revisões periódicas de regras. Essas revisões servem para garantir que o maior número possível de cidadãos tenham acesso aos programas e que esse acesso possa ser contínuo.

Nestas revisões, os sistemas computacionais que suportam os programas costumam sofrer mudanças com a adição ou modificação de módulos ou até mesmo a criação de novos sistemas, podendo, ou não, levar à descontinuação dos sistemas anteriores, agora ultrapassados.

Estes novos sistemas e módulos acabam implementando novas tecnologias e populando novos bancos de dados. Por isso, é difícil criar indicadores para programas governamentais. Os dados são provenientes de bancos de dados criados por pessoas que talvez não estejam mais no órgão responsável pelo programa, ou que trabalham em empresas terceirizadas cujos contratos já terminaram.

Para mitigar esse problema, é proposto um novo sistema, que tem como objetivo integrar os bancos de dados das aplicações utilizadas nesses programas governamentais, a fim de criar indicadores sobre estes programas.

Nesse novo sistema, todos os bancos de dados são integrados em um único, a fim facilitar a criação de um sistema com indicadores de um programa governamental.



Como desenvolver um sistema desse tipo não é trivial, a experiência dos autores no desenvolvimento é relatada.

Portanto, a principal contribuição deste trabalho é o relato de como foi desenvolver um sistema que unifica bases de dados periodicamente e permite a criação de indicadores sobre programas governamentais de grande impacto. Outra contribuição desse trabalho é a arquitetura implementada, que permite a criação de indicadores e os atualiza regularmente.

Então, neste trabalho são apresentados: a definição do problema na Seção II, os trabalhos relacionados na Seção III, a arquitetura do sistema criado na Seção IV, o fluxo de desenvolvimento da solução é descrito na Seção V, os problemas encontrados na Seção VI, como esses problemas foram resolvidos na Seção VII e uma discussão e conclusão deste trabalho apresentado na Seção VIII.

## II. DEFINIÇÃO DO PROBLEMA

Como descrito anteriormente, na Seção I, o sistema a ser desenvolvido precisa ser capaz de sincronizar múltiplos bancos de dados para consolidar indicadores sobre programas governamentais. O sistema resultante deve permitir que todos os cidadãos possam acessar indicadores sobre o programa em questão, sejam eles engajados com o programa ou não. Portanto, é necessário desenvolver um sistema de permissões para que os indicadores com dados sensíveis possam ser vistos apenas por usuários com aprovação para tal.

Como os dados para a criação dos indicadores são provenientes de outros sistemas em produção, o sistema precisa ser capaz de sincronizar os dados periodicamente de forma automática e manualmente, quando um administrador quiser, de forma a permitir a exibição de dados históricos e atualizados.

Os indicadores criados também devem poder ser baixados no formato CSV, permitindo o uso de outras ferramentas de análise de dados, como softwares de planilhas eletrônicas, e linguagens de programação, como Python e R.

## III. TRABALHOS RELACIONADOS

Nesta Seção são apresentados os trabalhos relacionados ao projeto desenvolvido e a esse artigo. Na Subseção III-A outros trabalhos de relato de experiência são apresentados. A Subseção III-B apresenta os conceitos de *Business Intelligence (BI)* que nortearam o desenvolvimento desse trabalho. As ferramentas de gestão de dados que serviram como base durante a concepção desse trabalho são descritas brevemente na Subseção III-C.

### A. Relato de experiência

Relato de experiência em desenvolvimento de software é uma prática comum, já que estes trabalhos costumam ajudar na definição de técnicas para a execução de novos projetos.

Neste sentido, Aragão et al. [1] relatam a modelagem de um processo de desenvolvimento de software que aconteceu de forma iterativa, envolvendo todos os membros da equipe de desenvolvimento. O processo modelado auxiliou a equipe a ter uma visão geral das atividades executadas e também na integração de novos membros.

Rivero et al. [2] reportam o desenvolvimento de um sistema baseado em *machine learning* para tentar prever quais clientes estão insatisfeitos com os serviços de uma empresa. Para tanto a equipe utilizou a técnica de desenvolvimento *gitflow*. Nesta técnica, utiliza-se de um software de gestão de configuração chamado *git* para sincronizar o desenvolvimento da aplicação.

Portilho et al. [3] relatam a experiência de desenvolver uma aplicação de feira eletrônica utilizando Delphi com alunos do ensino médio em um programa integrador multidisciplinar. Os resultados obtidos nesse projeto ajudaram os feirantes a comercializar seus produtos durante a pandemia da COVID-19.

### B. Business Intelligence

*Business Intelligence (BI)* utiliza um conjunto de tecnologias para suportar a tomada de decisões [4], [5]. Para suportar o *BI*, geralmente utiliza-se de uma técnica chamada *data warehouse*, em tradução livre armazém de dados, pela qual são reunidos todos os dados de uma instituição, além de todas as transformações por eles sofridas, resultado dos tratamentos desses dados e que geralmente formam os chamados cubos de dados. Esses cubos se caracterizam como tabelas que contém apenas informações relevantes para resolver problemas específicos de cada equipe de uma instituição [6], [7].

Então, por exemplo, em uma rede de supermercados, um cubo possível é a quantidade de produtos vendidos. Esta informação será armazenada com múltiplas dimensões: por marca, por dia, por mês, por ano, por cliente, por loja, por tipo de produto e por meio de venda (aplicativo, site, loja física). Dessa forma, um gestor poderá consultar quantos produtos foram vendidos agregando ou não por uma ou mais dimensões presentes no cubo de dados.

No que diz respeito à arquitetura de *BI*, Dayal et al. [8] estudaram dois tipos de arquitetura de *BI*. A primeira é a tradicional, que coleta dados de diferentes bancos de dados operacionais e dados externos. Esses dados passam por uma etapa de ETL (*Extract, Transform, Load*) que os integra para gerar um banco de dados de *data warehouse*. A partir desse banco de dados de *data warehouse*, um gerenciador de banco de dados pode criar muitos cubos de dados para atender a uma ampla variedade de propósitos. Em contraste, a próxima geração de inteligência de negócios transforma a etapa ETL em uma etapa bidirecional, ou seja, em vez de apenas tratar os

dados operacionais para o armazenar no *data warehouse*, esses dados tratados também são salvos como dados operacionais.

Mukherjee e Kar [9] investigaram ferramentas populares de integração de dados. Seu trabalho comparou os softwares Informatica, Datasatage, Ab Initio, SSIS e ODI em diversas categorias. No entanto, seu uso em projetos governamentais pode não funcionar porque todos eles são softwares proprietários que exigem a compra de licença, ou uma assinatura para serem utilizados.

Muryjas et al. [10] propuseram um processo de avaliação de professores e cursos durante a pandemia de COVID-19. A abordagem é semelhante à proposta por Villegas-Ch et al. [11]. Muryjas et al. [10] aplicaram o conceito de cubos de dados e *business intelligence* para atingir seus objetivos. Em seu processo, eles coletam dados de várias fontes, os transformam e integram. Em seguida, eles criam cubos de dados para construir diferentes painéis contendo informações para diferentes níveis de acesso.

### C. Ferramentas

Ehrenfried et al. [12] desenvolveram uma ferramenta para melhorar a fase de coleta de dados de um processo de inteligência de negócios orientado a dados abertos. Em seu trabalho, o método proposto auxilia na evolução dos dados, definindo um protocolo de mapeamento para cada ano. Esse protocolo trata todas as diferenças entre os anos, de modo que a tabela resultante é padronizada.

Villegas-Ch et al. [11] criaram uma ferramenta para analisar dados educacionais durante a pandemia da COVID-19 para melhorar os resultados de aprendizagem. Em seu trabalho, foram tomados como base os dados de informações dos alunos, e as informações foram complementadas com outras fontes de dados. Todos esses dados são processados, com o empenho de técnicas de armazenamento de dados: como ETL, a criação de cubos e a análise dos dados - usando técnicas de mineração de dados. Desses resultados são criados painéis de controle para apoiar decisões sobre o estado de aprendizagem dos alunos.

## IV. ARQUITETURA DO SISTEMA

O sistema desenvolvido foi pensado a fim de permitir o reuso por outros programas governamentais. Portanto, ele é um sistema modular. O sistema é composto de quatro módulos principais: O banco de dados, um backend administrativo, um backend de indicadores e o frontend. Uma visão geral do sistema é apresentada na Figura 1.

Detalhes sobre cada módulo apresentado na Figura 1 são descritos nas Subseções a seguir.

Esses módulos implementam um sistema que tem cinco níveis de permissões, cada usuário cadastrado no sistema tem acesso a um dashboard particular que pode ser personalizado ao

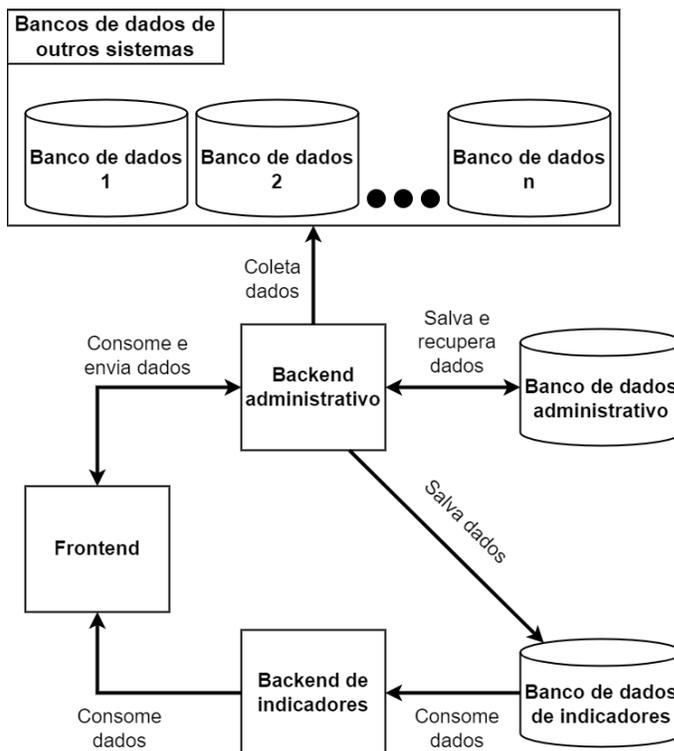


Fig. 1. Visão geral da arquitetura do sistema

seu gosto. Entretanto, mesmo usuários não cadastrados podem ver indicadores do sistema, por meio de um dashboard público, cuja customização só pode ser feita por um administrador do sistema. Nesses dashboards são exibidos indicadores que retratam o estado de um programa governamental. Além disso, esse sistema conta com um sistema administrativo que permite a gerência dos indicadores e a gerência de acesso dos usuários a indicadores com dados sensíveis.

### A. Banco de dados

O sistema gerenciador de banco de dados escolhido é o PostgreSQL, uma vez que é um sistema de software livre com uma comunidade ativa. Além disso, o PostgreSQL suporta diversas extensões. Entre estas extensões, as que são utilizadas nesse projeto são a PostGIS, para armazenamento de dados geográficos, e a Citus Data que, entre outras funcionalidades, permite a transformação da forma de armazenamento dos dados das tabelas de linear para colunar. A diferença entre uma tabela linear para a colunar é a forma de armazenamento dos dados. Na tabela linear a tupla é armazenada em sequência na memória, enquanto que na tabela colunar, cada coluna é armazenada sequencialmente.



Ambos os bancos de dados apresentados na Figura 1 são gerenciados pelo PostgreSQL. Entretanto, cada banco de dados tem configurações diferentes, descritas a seguir.

O banco de dados de indicadores utiliza o armazenamento colunar nas tabelas que armazenam os cubos de dados. Ao utilizar o sistema de armazenamento colunar, podemos melhorar o desempenho de consultas analíticas, como as consultas executadas nestes cubos de dados. Como as informações estão em sequência na memória, a busca das informações de uma coluna é mais simples, e portanto o desempenho de funções de agregação como contagem, soma e média são melhorados [13].

O outro banco de dados do sistema, o banco de dados administrativo, não utiliza nenhum módulo adicional do PostgreSQL. Ele foi pensado para comportar os dados que serão usados pelo frontend, com sua manipulação feita pelo backend administrativo. Nesse banco de dados são guardadas informações sobre dashboards, usuários, indicadores e permissões de usuários.

### B. Backend administrativo

O backend administrativo é um sistema escrito usando Loop-Back 4, que fornece uma API REST cujo banco de dados é armazenado no PostgreSQL. Com esse sistema, é possível permitir o login de usuários, recuperar dados de indicadores, editar dashboards particulares e registrar novos indicadores - para citar algumas de suas funcionalidades.

Além das funções já descritas, o backend administrativo também é responsável por sincronizar os dados entre todos os bancos de dados que contenham informações que podem virar indicadores. Para tanto foi desenvolvido um submódulo, escrito em Python, que coleta os dados de todos os bancos de dados e cria os cubos de dados no banco de dados de indicadores.

A escolha por implementar esse submódulo em Python veio da alta disponibilidade de adaptadores de bancos de dados, desta forma caso exista um novo banco de dados para fornecer dados para indicadores, bastará um desenvolvedor implementar as funções básicas do sistema utilizando o adaptador.

Esse módulo foi implementado nesse backend, porque um dos requisitos desse painel de indicadores é a possibilidade de sincronizar os dados dos indicadores. Essa sincronia de dados poderá ser realizada automaticamente ou a qualquer momento por um administrador e, para garantir que o usuário que executa essa função é um administrador, é necessário o sistema administrativo que esse backend implementa.

### C. Backend de indicadores

O backend de indicadores é um sistema implementado em Python usando o framework `fastapi` para criar uma API HTTP que disponibiliza os dados dos indicadores do sistema.

Como os indicadores do sistema são armazenados em cubos de dados, é responsabilidade do backend de indicadores

permitir o acesso aos cubos de dados. Para tanto, é criada uma consulta que retorna os dados sem dimensão alguma. Essa consulta pode ser modificada adicionando dimensões que mudam a granularidade da resposta retornada.

Usando o exemplo fornecido na Seção IV-A no qual existe uma rota que retorna a quantidade de produtos vendidos por uma rede de supermercados, esta rota pode ser chamada com a função GET como é mostrado a seguir:

```
GET api.mercado/v1/qt_prod
```

O retorno desta chamada pode ser algo como:

```
{
  resultado: [
    {
      total_prod_vendido: 99651980000
    }
  ]
}
```

Caso o usuário queira ver quantos produtos são vendidos por ano, mês e dia, ele pode fazer uma chamada para esta rota da seguinte forma:

```
GET api.mercado/v1/qt_prod?dims=dia,mes,ano
```

O que resultaria em uma resposta similar a seguinte:

```
{
  resultado: [
    ...
    {
      ano: 2023,
      mes: "jan",
      dia: 1,
      total_prod_vendido: 52364
    },
    {
      ano: 2023,
      mes: "jan",
      dia: 2,
      total_prod_vendido: 124334
    },
    ...
    {
      ano: 2023,
      mes: "jun",
      dia: 30,
      total_prod_vendido: 96425
    }
  ]
}
```

É possível observar que, se todos os campos `total_prod_vendido` dessa consulta agregada por dia forem somados, o valor resultante é o valor da resposta da API sem nenhuma dimensão, ou seja 99651980000.

Esse backend também permite o usuário realizar o download dos dados em outros formatos, como em CSV, por exemplo. Desta forma, o requisito de disponibilização de dados é cumprido.

#### D. Frontend

Frontend é o sistema responsável pela interação entre usuários e dados. Para implementar o frontend desse projeto foi utilizado o React, um framework de desenvolvimento web. Nesse framework, a equipe responsável pelo desenvolvimento implementou a interface seguindo o padrão digital do governo brasileiro, o DSGOV<sup>1</sup> (*Design System do Governo*).

Essa interface recebe os dados dos dois backends. As informações do backend de indicadores são utilizadas para plotar tabelas e gráficos que compõem o dashboard. Esse dashboard pode ser personalizado, permitindo um melhor controle sobre as informações do programa governamental. É possível escolher mostrar indicadores em linhas completas ou em meia linha. Pode-se também pular espaços, assim como exemplifica a Figura 2

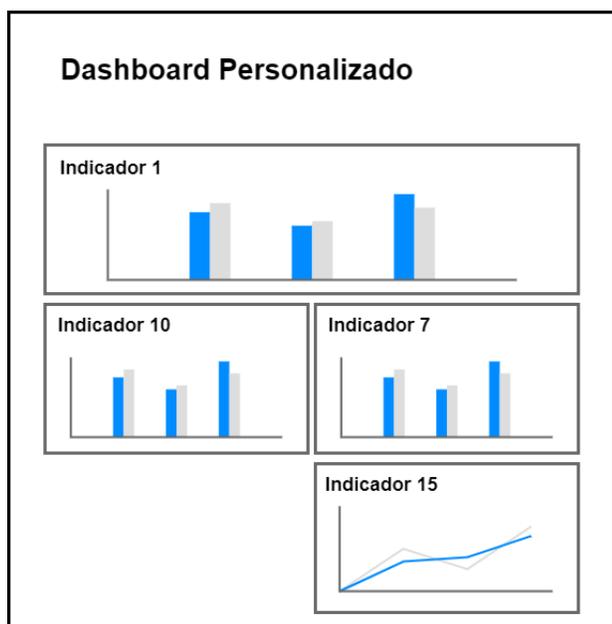


Fig. 2. Exemplo de Dashboard personalizado

A utilização do backend administrativo permite a criação de usuários, a gerência de permissões de cada usuário, a gerência dos indicadores e o salvamento do indicador personalizado de cada usuário. Todas estas funcionalidades, além de outras, podem ser acessadas pelo frontend do sistema.

#### V. FLUXO DE DESENVOLVIMENTO

Para ser capaz de realizar o desenvolvimento desse sistema, a equipe de desenvolvedores foi dividida em três grupos: 1) Equipe de banco de dados (EBD), 2) Equipe do frontend (EF)

<sup>1</sup><https://www.gov.br/ds/home> - Último acesso em 04 de outubro de 2023

e 3) Equipe do backend (EB). Todos os grupos utilizaram o fluxo de trabalho *gitflow*.

A EBD ficou inicialmente responsável por modelar o banco de dados administrativo, já que a EB precisava desta modelagem para o backend administrativo. Enquanto o banco de dados administrativo não estava modelado, a EB desenvolvia a estrutura do backend de indicadores para suportar a consulta dos cubos de dados. Ao mesmo tempo, a EF desenvolvia todas as telas estáticas do frontend.

Quando a EBD terminou de modelar o banco de dados administrativo, ela passou para a tarefa de examinar os bancos de dados dos outros sistemas e a criar a definição dos cubos de dados. Da mesma forma, quando a EB terminou de implementar a base do backend de indicadores, ela passou a implementar o backend administrativo ao mesmo tempo que a EF desenvolvia as telas que usam esse backend (exemplo: login, recuperar senha, gerenciamento de indicadores, entre outros).

Em sequência, a EB começou a implementar as APIs para cada cubo de dado ao mesmo tempo que a EF desenvolvia as telas que usavam esses cubos de dados. Um esquema das tarefas desenvolvidas no projeto é mostrada na Figura 3.

É possível observar que as tarefas ilustradas na Figura 3 em sua maioria possuem dependências entre tarefas de equipes diferentes. Portanto, foi preciso organizar as tarefas de modo que elas acontecessem, no pior caso, ao mesmo tempo. Dessa forma, foi possível implementar o sistema eficientemente.

Outra observação é que, no final do fluxo da Figura 3, a EBD se transforma em equipe de cientistas de dados, já que a tarefa que os membros precisam fazer diz respeito a investigar os dados e criar cubos de dados.

Durante o desenvolvimento, todas as dúvidas foram salvas em documentos e encaminhadas para os coordenadores do projeto, e estes encaminhavam as dúvidas a quem conseguisse responder. Quando a maioria das dúvidas eram respondidas, um documento com as respostas era encaminhado para a equipe de desenvolvimento.

#### VI. PROBLEMAS ENCONTRADOS

Por utilizar dados armazenados em bancos de dados do governo federal do Brasil, conseguir acesso a esses dados é uma tarefa burocrática. Esses dados estão sob responsabilidade de equipes diferentes, que possuem diferentes protocolos para sua disponibilização. Alguns dos protocolos necessários para obter acesso aos dados levam meses para serem processados.

Quando o acesso aos dados é conseguido, um novo problema surge: sua compreensão. Das bases de dados consultadas, nenhuma delas possui dicionário de dados, fato que dificulta e reduz a produtividade do desenvolvimento. Desta forma foi preciso investigar todo o banco de dados cuidadosamente para

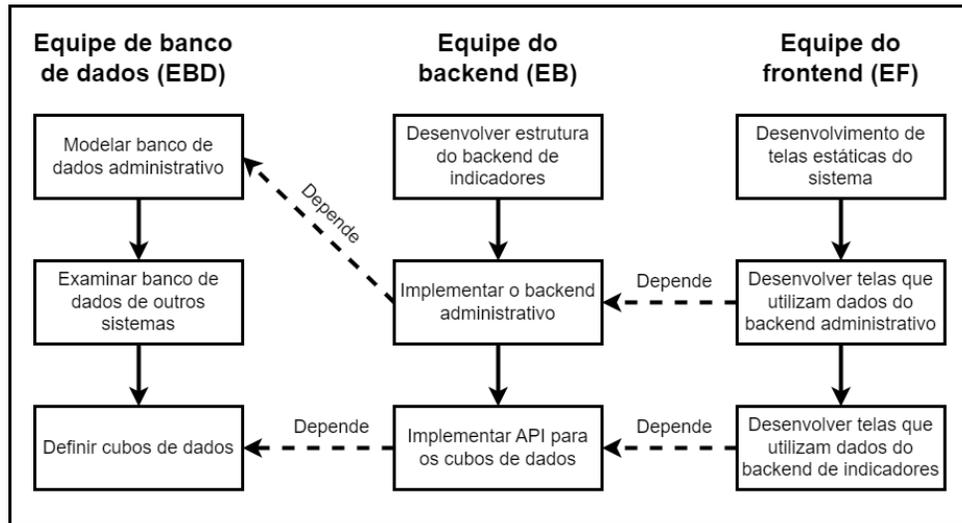


Fig. 3. Tarefas executadas no projeto. As linhas contínuas indicam sequência de tarefas e as linhas tracejadas indicam dependência entre tarefas.

entender o que cada tabela armazena, e como funciona a semântica dos dados em cada tabela.

Além disso, os desenvolvedores originais das bases de dados não estão mais nas equipes de desenvolvimento, e os membros atuais destas equipes não são especialistas na arquitetura desses bancos de dados, portanto nem sempre eles conseguem ajudar no entendimento do banco de dados.

Diante desse cenário, foi preciso realizar uma etapa adicional ao projeto, rodadas de entrevistas não estruturadas entre as equipes de *stakeholders* responsáveis pelos sistemas, para que fosse possível mapear questões que geraram dificuldades de entendimento quando do acesso aos bancos, tais como nomenclaturas e histórico de armazenamento dos dados. Essas pesquisas resultaram na identificação de diversas informações indispensáveis ao manuseio e tratamento dos dados, mas também apontou para conflitos entre os termos apresentados nos bancos disponíveis, o que poderia dificultar o planejamento do projeto.

Portanto, identificar os dados no banco de dados torna-se uma tarefa ainda mais complexa, pois necessita envolver cientistas de dados e colaboradores do governo federal. Essa observação se torna um problema porque, em geral, esses programas governamentais têm mais demanda do que colaboradores. Cada colaborador do programa que é consultado para desenvolver o sistema é especialista no âmbito de uma fase específica do programa. Por isso, não é possível ter ao lado dos cientistas de dados, durante todo o tempo, pessoas que entendem completamente sobre o programa governamental, o que gera atrasos no desenvolvimento de indicadores semanticamente corretos.

## VII. SOLUÇÃO DOS PROBLEMAS

A questão de burocracia teve que ser superada com paciência e proatividade. A equipe provia toda a documentação antes mesmo de ser requisitada, e dessa forma foi possível reduzir um pouco o tempo necessário para conseguir o acesso aos dados.

Já sobre o fato de não ter dicionário de dados, a equipe investigou cada tabela, a fim de identificar as tabelas mais relevantes para a criação de indicadores. Entretanto, como ninguém que entende sobre os dados estava junto com a equipe durante essa etapa, é possível que nem todas as tabelas realmente necessárias tenham sido identificadas corretamente.

Por fim, a respeito da dificuldade em ter as informações dos colaboradores que entendem as regras do programa, a equipe de cientistas de dados criava listas de perguntas e as encaminhava periodicamente para a instituição responsável pelo programa governamental. Com esse material, também foram realizadas reuniões para discutir algumas dúvidas pontuais sobre os dados.

## VIII. CONCLUSÃO

Nesse artigo foi apresentada a arquitetura de um sistema desenvolvido para permitir a criação de indicadores para programas governamentais, e o relato da experiência em desenvolver esta arquitetura. O sistema desenvolvido seguindo essa arquitetura permite a personalização do sistema e o reuso em outros cenários.

Também foram discutidos os problemas encontrados, e como a equipe de desenvolvimento os superou durante o desenvolvimento do projeto.

É possível identificar que, com exceção da burocracia, fato que não é necessariamente ruim, os outros problemas en-



contrados vieram de duas fontes: 1) a equipe que entende do negócio estava muito atarefada e não conseguia realizar reuniões periódicas para o desenvolvimento do sistema; 2) a equipe que desenvolveu o sistema original não o documentou, e utilizou nomenclatura inconsistente com a usada pela equipe técnica do governo.

Considerando o desenvolvimento, o ponto 2 foi o que mais afetou o desenvolvimento do sistema, já que demandou que a equipe aprendesse mais sobre os dados mesmo com pouco suporte na investigação.

Ainda assim, o trabalho realizado pode ser considerado um sucesso, pois um dos resultados obtidos é o desenvolvimento de uma arquitetura reutilizável que permite a criação de dashboards públicos e particulares. Além disso, com os dados obtidos e tratados até então foi possível criar alguns indicadores para um programa governamental.

Em trabalhos futuros será investigado como a arquitetura desenvolvida funcionaria com outro projeto. Também será analisado questões de Interação Humano Computador, para garantir que a interface desenvolvida seja o mais acessível possível.

Também é possível incluir nos trabalhos futuros a criação simplificada de novos indicadores, já que atualmente cada indicador precisa ser desenvolvido por um desenvolvedor de software.

#### AGRADECIMENTOS

Pesquisa financiada com recursos do Fundo Nacional de Desenvolvimento (FNDE) e Ministério da Educação (MEC) do Governo Federal do Brasil, com execução realizada pelo Centro de Computação Científica e Software Livre (C3SL) da Universidade Federal do Paraná (UFPR).

#### REFERÊNCIAS

- [1] B. Aragão, I. Santos, T. Nogueira, L. Mesquita, and R. Andrade, "Modelagem interativa de um processo de desenvolvimento com base na percepção da equipe: Um relato de experiência," in *Anais do XIII Simpósio Brasileiro de Sistemas de Informação*. Porto Alegre, RS, Brasil: SBC, 2017, pp. 428–435. [Online]. Available: <https://sol.sbc.org.br/index.php/sbsi/article/view/6072>
- [2] L. Rivero, I. Silva, P. Cutrim, A. Paiva, M. Oliveira, E. Alves, G. B. Junior, and D. Dias, "Implementando o gitflow para gerencia de configuração em um projeto de desenvolvimento de software ágil: Um relato de experiência," *Anais do Computer on the Beach*, vol. 12, pp. 178–185, 2021.
- [3] F. J. Portilho, E. T. G. Grande, K. C. Lima, and L. V. B. C. Leite, "Experience report from an integrated project developing a street food market managing software in iporá," *Research, Society and Development*, vol. 10, no. 1, p. e1281011283, Jan. 2021. [Online]. Available: <https://rsdjournal.org/index.php/rsd/article/view/11283>
- [4] S. Chaudhuri and U. Dayal, "An overview of data warehousing and olap technology," *SIGMOD Rec.*, vol. 26, no. 1, p. 65–74, 03 1997. [Online]. Available: <https://doi.org/10.1145/248603.248616>
- [5] L. Cabibbo and R. Torlone, "A logical approach to multidimensional databases," in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1998, pp. 183–197. [Online]. Available: <https://doi.org/10.1007/bfb0100985>
- [6] A. Bonifati, F. Cattaneo, S. Ceri, A. Fuggetta, and S. Paraboschi, "Designing data marts for data warehouses," *ACM Transactions on Software Engineering and Methodology*, vol. 10, no. 4, pp. 452–483, Oct. 2001. [Online]. Available: <https://doi.org/10.1145/384189.384190>
- [7] I.-Y. Song, *Data Mart*. Boston, MA: Springer US, 2009, pp. 594–594. [Online]. Available: [https://doi.org/10.1007/978-0-387-39940-9\\_883](https://doi.org/10.1007/978-0-387-39940-9_883)
- [8] U. Dayal, M. Castellanos, A. Simitsis, and K. Wilkinson, "Data integration flows for business intelligence," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, ser. EDBT '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 1–11. [Online]. Available: <https://doi.org/10.1145/1516360.1516362>
- [9] R. Mukherjee and P. Kar, "A comparative review of data warehousing etl tools with new trends and industry insight," in *2017 IEEE 7th International Advance Computing Conference (IACC)*, 2017, pp. 943–948.
- [10] P. Murjas, M. Wawer, and M. Rzemieniak, "Managing the process of evaluation of the academic teachers with the use of data mart and business intelligence," *EUROPEAN RESEARCH STUDIES JOURNAL*, vol. XXIV, no. Special Issue 2, pp. 127–140, Jun. 2021. [Online]. Available: <https://doi.org/10.35808/ersj/2196>
- [11] W. Villegas-Ch, X. Palacios-Pacheco, and S. Luján-Mora, "A business intelligence framework for analyzing educational data," *Sustainability*, vol. 12, no. 14, 2020. [Online]. Available: <https://www.mdpi.com/2071-1050/12/14/5745>
- [12] H. V. Ehrenfried, E. Todt, D. Weingaertner, L. C. E. D. Bona, F. Silva, and M. D. D. Fabro, "Managing open data evolution through bi-dimensional mappings," in *Proceedings of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, BDCAT 2019, Auckland, New Zealand, December 2-5, 2019*, K. Johnson, J. Spillner, X. Zhao, O. Datskova, and B. Varghese, Eds. ACM, 2019, pp. 159–162. [Online]. Available: <https://doi.org/10.1145/3365109.3368774>
- [13] D. J. Abadi, S. R. Madden, and N. Hachem, "Column-stores vs. row-stores: how different are they really?" in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 967–980.

