

Desenvolvimento de uma função de mapeamento relacional para processamento automatizado de relatórios de inspeção da indústria extrativa

Bruno Lobell* Lucas Marca* Junior Zilles* Eder Mateus Gonçalves*
Emanuel Estrada* Eduardo N. Borges* Danúbia B. Espíndola*
Sílvia S. da Costa Botelho*

* Centro de Ciências Computacionais, Universidade Federal do Rio Grande, RS, (e-mail: {brunolobell, lucasmarca, juniorzilles, edergoncalves, emanuelestrada, eduardoborges, dmtdbe, silviacb}@furg.br).

Abstract: The inspection process is one of the most important in the extractive industry. Among the activities of this process, the evaluation of inspection reports stands out. The reports attest whether a given equipment meets the minimum requirements for its use, usually defined by manufacturing standards defined by regulatory institutions. Consulting, checking and analyzing these printed reports is a very costly procedure. The present work presents a method and a prototype that implements the mapping of the result of pipe inspection reports in construction and assembly databooks of the extractive industry in a relational data model. The automatically extracted structured data models contributed significantly to the increase in quality and selectivity of industrial inspection review processes.

Resumo: O processo de inspeção é um dos mais importantes da indústria extrativa. Entre as atividades deste processo destaca-se a avaliação de relatórios de inspeção. Os relatórios atestam se um determinado equipamento atende aos requisitos mínimos para sua utilização, usualmente definidos por normas de fabricação definidas por instituições regulamentadoras. A consulta, verificação e análise desses relatórios impressos é um procedimento muito custoso. O presente trabalho apresenta um método e um protótipo que implementa o mapeamento do resultado de relatórios de inspeção de tubulação em databooks de construção e montagem da indústria extrativa em um modelo relacional de dados. Os modelos de dados estruturados extraídos automaticamente contribuíram significativamente para o incremento de qualidade e a seletividade dos processos de revisão da inspeção industrial.

Keywords: Inspection; Pipe; Shipbuilding; Document processing.

Palavras-chaves: Inspeção; Tubulação; Construção naval; Processamento de documentos.

1. INTRODUÇÃO

A construção naval possui diversos procedimentos que garantem a qualidade da execução realizada e um deles é a inspeção naval, definida no país pela Lei 9.537/1997 como uma atividade de cunho administrativo. Esta atividade consiste na fiscalização do cumprimento desta Lei, das normas e dos regulamentos dela decorrente, e dos atos e resoluções internacionais ratificados pelo Brasil, no que se refere exclusivamente à salvaguarda da vida humana e à segurança da navegação, no mar aberto e em hidrovias interiores, e à prevenção da poluição ambiental por parte de embarcações, plataformas fixas ou suas instalações de apoio, Brasil (1997). Logo, uma construção naval para ter liberação de funcionamento deve passar por testes que garantem a sua qualidade.

Os testes de qualidade aplicados sobre uma construção naval são efetuados inicialmente pelo fornecedor de componentes e em seguida por um inspetor qualificado para a categoria de equipamento. As inspeções são registradas em

documentos comprobatórios de que o equipamento passou ou não pela inspeção, sendo no final assinadas e carimbadas pelo inspetor envolvido e responsáveis técnicos. Os documentos são chamados relatórios de inspeção e seu *layout* pode variar conforme a inspeção realizada. Alguns exemplos de relatórios de inspeção incluem Dimensional de Ajuste, Dimensional de Campo, Visual de Ajuste e Visual de Solda. Por fim, o documento final é um montante de folhas de papel que deve ser armazenado para consultas posteriores.

A Lei 9.537/1997 também define plataforma como uma instalação ou estrutura, fixa ou flutuante, destinada às atividades direta ou indiretamente relacionadas com a pesquisa, exploração e exploração dos recursos oriundos do leito das águas interiores e seu subsolo ou do mar, inclusive da plataforma continental e seu subsolo, Brasil (1997). As plataformas comportam tubulações dos mais diferentes tipos e com diferentes funcionalidades, destinadas a líquidos e/ou gases, na maioria inflamáveis e sob alta

pressão, sendo por isso um dos pontos mais importantes na construção foca das inspeções navais.

Os relatórios de inspeção são documentos impressos. Mesmo quando digitalizados e armazenados em formato aberto como PDF, não permitem a consulta automatizada por informações de interesse, porque são armazenados como imagem. O presente trabalho apresenta um método e implementa uma solução para extração automatizada de informação textual relevante de relatórios de inspeção. Nesta primeira etapa a extração é baseada no Reconhecimento Óptico de Caracteres (OCR) e no armazenamento do conteúdo extraído em formato de dados semiestruturados. Após, os dados são interpretados e mapeados para um modelo relacional definido a priori, que descreve a estrutura dos relatórios de inspeção, permitindo realizar as mais diversas consultas sobre os dados.

O restante do texto está organizado da seguinte forma. Os trabalhos relacionados são sintetizados na Seção 2. Na Seção 3 é descrito o método proposto, a sua implementação como um serviço Web e as tecnologias utilizadas. Os resultados alcançados e um estudo de caso para compreensão do funcionamento e finalidade são retratados na Seção 4. Por fim, o trabalho é concluído na Seção 5.

2. TRABALHOS RELACIONADOS

O escopo definido para este artigo abrange o problema da serialização de dados provenientes da tarefa de inspeção de tubulações. Por ser um tópico muito específico, não foram encontrados trabalhos que tratem exatamente do escopo definido, mas foram investigados estudos que abordam serialização de dados e outros que tratam da inspeção de tubulações.

Roque (2020) apresenta a inspeção por ensaios não destrutivos (END) como técnica que permitem o controle das propriedades dos materiais, com economia de tempo e dinheiro, permitindo que o material testado volte intacto para o local de trabalho após a inspeção. O autor avaliou o resultado da inspeção de diferentes técnicas utilizadas nos ensaios. Costa e Silva (2017) discute a importância da manutenção preditiva dos dutos utilizando de técnicas ou END para o acompanhamento de parâmetros, antecipando o melhor momento para se efetuar a parada do equipamento para manutenção corretiva. O autor aponta também que as tubulações trabalham 24 horas por dia sete dias por semana corroborando com o argumento em questão da mínima interferência de sobre a tubulação.

A dissertação de Geremia (2012) aponta que a principal forma de transporte de petróleo e derivados, entre outros produtos, são os dutos. Um dos processos críticos relacionados é a sua inspeção periódica, necessária e obrigatória, além de ter alto custo devido à complexidade e a que na maioria das vezes é realizada manualmente, em procedimentos demorados, pouco confiáveis e insalubres. Para solucionar o problema, o autor propõe testar aplicações e situações de um sistema robótico de inspeção de dutos.

Silva et al. (2020) dissertam a respeito da inspeção radiográfica de juntas soldadas em tubulações de petróleo como essencial para prevenir possíveis vazamentos, que impactam em diferentes setores: econômico, ambiental e da saúde. Os inspetores analisam radiografias de juntas sol-

dadas para identificar defeitos como porosidades e trincas. O trabalho propõe filtrar a base de dados de treinamento usada para a detecção de falhas em juntas soldadas de tubulações. Esta base é construída a partir do resultado da avaliação e interpretação dos ensaios radiográficos de juntas soldadas na forma de laudos segundo os critérios de aceitação baseados em normas e especificações técnicas.

Diferentemente de Geremia (2012), que propõe a remoção do operador humano durante a realização da inspeção, o trabalho proposto neste artigo foca no processamento e conferência automatizados dos resultados da inspeção, ou seja, auxilia na tarefa mais custosa do inspetor. Os relatórios de inspeção dos END são entradas para o OCR e posterior processamento pelo sistema proposto. Quando comparado ao trabalho de Silva et al. (2020), flexibilizamos a abrangência para todas as categorias de relatórios de inspeção de tubulação, e não apenas raios-x.

Em relação à serialização de dados foram estudados vários trabalhos, dos quais se destacam os relacionados a seguir.

A definição da estrutura de dados vinculados em JSON é abordada por Kellogg et al. (2019). Pode-se observar a sintaxe leve do JSON-LD e inferir que um dos melhores métodos de serialização entre dados relacionais é através de uma estrutura JSON. São utilizadas regras e atribuições de objetos em suas propriedades, facilitando o entendimento e o acesso dos valores nos atributos.

Dähling et al. (2020) apresenta uma ferramenta para serialização, deserialização e comunicação entre *Web Ontology Language* (OWL) e a linguagem de programação Go (diferente de Go!). Go é uma linguagem C-like, mas com algumas ferramentas extras. Kubernetes e Docker, por exemplo, foram desenvolvidas em Go. Nesse trabalho, descreve-se um modelo de ontologia e utilização de dados entre estruturas distintas, onde pode ser ttl(Turtle) ou JSON-LD. A serialização e deserialização é aplicada na utilização de duas estruturas entre os dados a serem representados na ontologia.

Da, Zhongyue et al. (2017) apresentam o funcionamento de um sistema em que a tarefa a ser realizada é construir o modelo de dados semiestruturados JSON e gerar o modelo de tabela de dados relacional correspondente. Em seguida, processar os dados semiestruturados em dados estruturados e armazená-los no banco de dados.

Diferente de Dähling et al. (2020), o trabalho proposto no presente artigo envolve o mapeamento JSON para banco de dados relacional. O procedimento é similar ao aplicado por Da, Zhongyue et al. (2017), com a diferença em relação ao formato de arquivo de entrada. O sistema proposto recebe um PDF em que é extraída uma imagem e um JSON com dados do OCR.

3. METODOLOGIA

A partir de uma amostra considerável de relatórios de inspeção fornecido por uma empresa extrativista brasileira, foi realizado um levantamento de requisitos de dados que resultou no projeto de dados apresentado no diagrama relacional da figura 1. Cada retângulo corresponde a uma relação ou tabela de dados, composta um conjunto de atributos ou campo com categorias de dados associados.

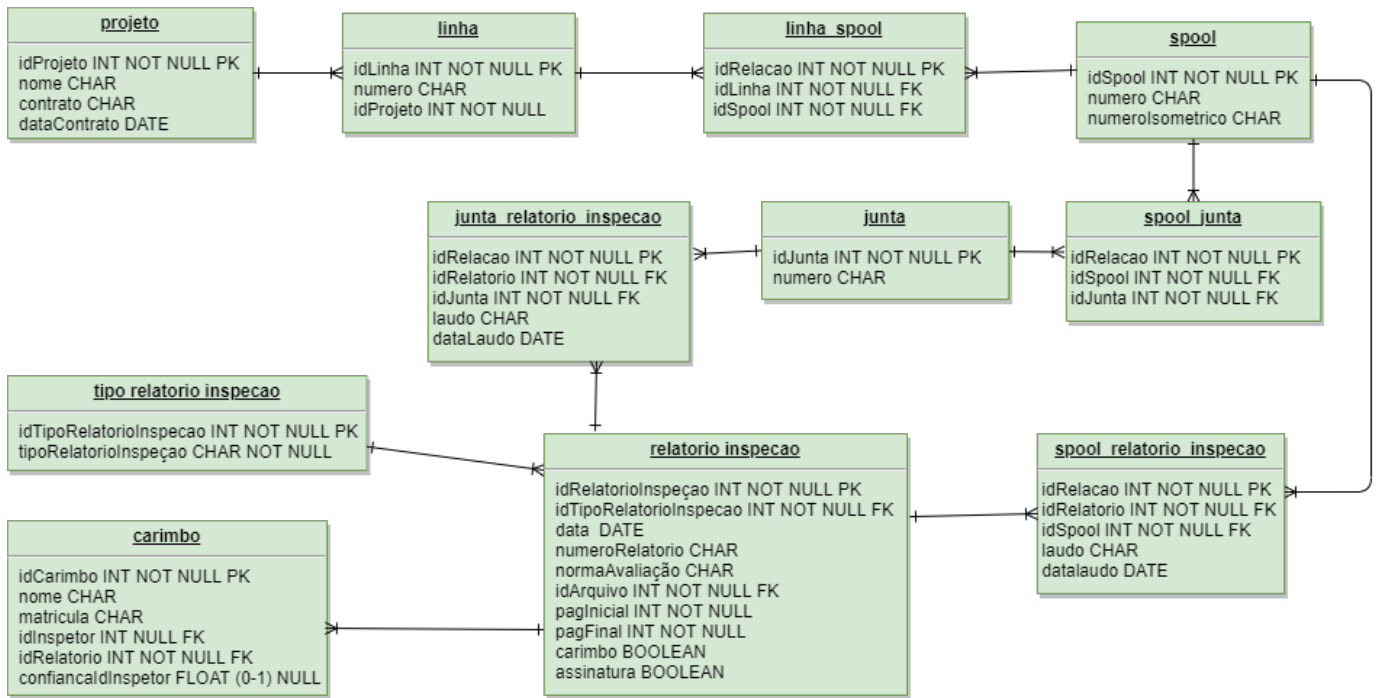


Figura 1. Projeto de dados relacional que define a estrutura das informações a serem armazenadas, levantadas na análise de requisitos.

Os relacionamentos entre as tabelas são representados pelas linhas e a cardinalidade pela notação pés de galinha Elmasri, Ramez and Navathe, Shamkant B. (2018).

Um projeto, ou seja, a construção e montagem da tubulação de uma plataforma é representada por um nome, um contrato e uma data de assinatura. Cada projeto engloba um conjunto de vários linhas, representadas apenas por um número. Por padrão, uma linha é constituída de um ou mais spools, que correspondem a tubos soldados a uma ou mais conexões tais como curvas, tês, reduções, flanges, ou luvas. Os spools são usados na montagem dos sistemas de tubulações e podem conectar mais de uma linha. Esse relacionamento M:N é mapeado pela tabela *linha_spool*.

Cada conexão de solda entre dois ou mais tubos, ou spools é denominada junta. Analogamente, *spool_junta* implementa o relacionamento M:N. Dos relatórios de inspeção são armazenadas informações sobre data, número, norma, identificador do arquivo PDF, página inicial e final, se possui carimbo ou assinatura. Cada junta com inspeção obrigatória deve estar contida num relatório contendo assinatura e carimbo do profissional responsável pela inspeção. A existência do profissional de inspeção e a adição de mais dois profissionais técnicos responsáveis pela avaliação do equipamento antes da sua aplicação em um ambiente de produção gera um relacionamento 1:N entre as entidades *relatorio_inspecao* e *carimbo*. Adicionalmente, um relatório de inspeção é caracterizado por um dos seguintes tipos: Dimensional de Ajuste, Dimensional de Campo, Visual de Ajuste, Visual de Solda, Partícula Magnética, Líquido Penetrante, Gamagrafia, Dureza, Ferrita, Raio-X e Liberação de Juntas Roscadas.

O desenvolvimento do sistema se deu pelo levantamento de requisitos de funcionamento aplicados à arquitetura de microsserviços. No presente caso, o processamento é

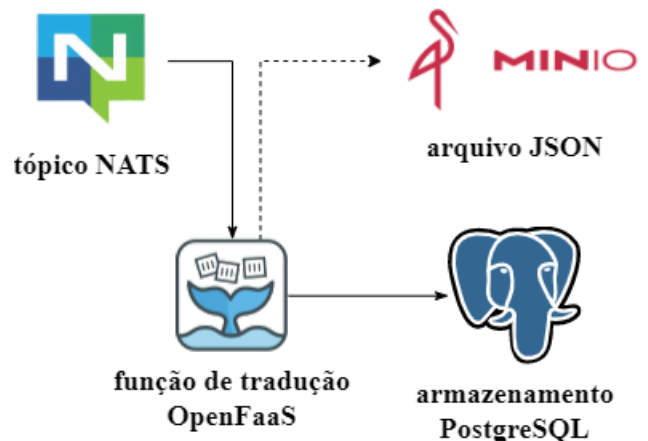


Figura 2. Arquitetura do método proposto para mapeamento dos relatórios de inspeção.

realizado sem a interação direta do usuário, visto que ele realiza apenas o upload do arquivo e o sistema realiza o processamento a partir dessa entrada. Os processamentos geram uma chamada no serviço de mensageria *NATS*¹ que disparam a execução do próximo processamento. Arquivos ou quaisquer outros artefatos intermediários são armazenados usando outro microsserviço denominado *MinIO*². Este sistema é um gerenciador de objetos digitais e facilita o armazenamento e recuperação de arquivos. Os processos também interagem com uma base de dados relacional, logo o escolhido foi o *PostgreSQL*³. O motivo desta escolha foi o conjunto de funcionalidades baseadas em múltiplas funções de similaridade textual. Por fim, o processamento

¹ <https://nats.io/>

² <https://min.io/>

³ <https://www.postgresql.org/>

```
1 {
2   "pages": [
3     {
4       "page_num": 1,
5       "class": 0,
6       "project_name": "projeto A",
7       "class_name": "N381_A1",
8       "class_prob": 80.0,
9       "stamp": true,
10      "signature": true,
11      "blocks": [
12        {
13          "id": "CLIENTE_Q",
14          "type": "form_question",
15          "bbox": [
16            535.0,
17            232.0,
18            111.0,
19            34.0
20          ],
21          "text": "CLIENTE:",
22          "links": [],
23          "conf": 1.0
24        }
25      ]
26    }
27  ]
28 }
```

Figura 3. Documento JSON contendo os dados textuais extraídos pelo OCR de um relatório de inspeção.

é realizado utilizando uma função como serviço (FaaS), escolhido o *OpenFaaS*⁴ como tecnologia de desenvolvimento e a linguagem de programação *Python*⁵ na versão 3. Os componentes do sistema são executados utilizando contêineres *Docker*⁶ no sistema de orquestração de contêineres *Kubernetes*⁷.

A figura 2 apresenta as etapas de processamento do método proposto para mapear os relatórios no modelo relacional de dados. Após o processamento de um arquivo PDF pelo OCR e conversão do seu conteúdo para o formato JSON, um tópico do NATS é disparado informando a disponibilidade de uma nova carga de trabalho. Este tópico é escutado pelo serviço OpenFaaS o qual direciona a execução da função de tradução implementada. Esta função de tradução obtém o arquivo JSON gerado pelo processo anterior no MinIO e processa as páginas do arquivo PDF realizando o mapeamento do texto extraído pelo ORC para o modelo relacional de dados e finalmente escreve o resultado no banco de dados do PostgreSQL conforme a estrutura apresentada na Figura 1.

Um exemplo de documento processado, armazenado no formato semiestruturado JSON, é apresentado na figura 3. É possível observar que o JSON possui um elemento principal *pages* que corresponde a uma lista de páginas. Cada página apresenta um número de página (*page_num*), nome do projeto (*project_name*), nome da classe (*class_name*) representando a categoria de inspeção, indício de que foi assinado e carimbado *signature* e *stamp*). No que lhe con-

cerne, o elemento *blocks* agrega os blocos de texto retirados do documento PDF. É possível observar a ocorrência de uma identificação de qual campo o texto se refere (*id*), o tipo do campo (*type*), sua posição no documento (*bbox*), o texto (*text*) e a confiança da qualidade do OCR realizado (*conf*).

4. RESULTADOS

Os documentos PDFs que compõem a documentação de um projeto de tubulação contêm uma sequência de relatórios de inspeção, gerados por uma epcista e regulamentados por um ou mais inspetores. Estes relatórios têm um padrão para cada tipo. A figura 4 mostra um exemplo de relatório do tipo Líquido Penetrante. A extração de dados textuais pelo OCR gera um arquivo JSON com as informações dos campos de cada página. Este arquivo é submetido ao serviço de armazenamento de objetos MinIO. Após, um tópico disparado pelo serviço de mensageria NATS a função de tradução do OpenFaaS começa sua execução. Os dados textuais semiestruturados são lidos e processados, descobrindo-se sequências de linha, *spool* e *junta*. Em seguida, a função analisa os dados recebidos, para corrigir algumas discrepâncias em campos incorretamente extraídos pelo OCR com o uso de expressões regulares, que verificam se o dado corresponde com o padrão definido. Por exemplo, a expressão "\d+/\d+/\d+", sendo "\d+" um grupo de caracteres com pelo menos um número de zero a nove, que irá recuperar uma sequência de três números separados por barra (/), os quais devem ser convertidos para uma data no formato ISO "ano-mês-dia". Os campos que, após a verificação, possuam alguma irregularidade são corrigidos e caso existam campos nulos eles são descartados.

O serializador insere os dados extraídos e analisados no banco de dados PostgreSQL, seguindo uma sequência preliminarmente definida, tendo início na tabela *projeto* e finalizando na tabela *carimbo*. A figura 4 apresenta os campos que são adicionados às tabelas do banco identificados em cores distintas para facilitar a explicação a seguir. Campos em vermelho (nome e contrato) e a data presente no final do documento são mapeados para a tabela *projeto*. Em seguida, os campos em azul (relatório, norma) fornecem os dados para a tabela *relatorio*, em conjunto com dois valores booleanos, que indicam a presença do carimbo e da assinatura dos responsáveis pela obra. A categoria de relatório Líquido Penetrante é identificado e referenciado no banco de dados. As tabelas *linha*, *spool* e *junta* armazenam os dados coletados das áreas marcadas, respectivamente, nas cores verde, amarelo e ciano. As tabelas *spool_relatorio_inspecao* e *junta_relatorio_inspecao* armazenam o resultado da inspeção, ou seja, se o laudo é aprovado (A) ou reprovado (R) e a data do laudo, identificados na cor laranja. Finalizando, a última tabela preenchida é a *carimbo*, a qual contém os dados relacionados aos responsáveis pela inspeção e coordenação da obra.

O documento JSON mostrado pela figura 5 representa a inserção do exemplo da figura 4 no banco de dados, sendo as chaves do JSON correspondentes às tabelas do banco de dados e os itens internos suas colunas. Todas as tabelas contêm um ou mais campos, os quais caracterizam um determinado elemento, fazendo com que não ocorra repetições. Os dados para a tabela *projeto*, sendo o nome

⁴ <https://www.openfaas.com/>

⁵ <https://www.python.org/>

⁶ <https://www.docker.com/>

⁷ <https://kubernetes.io/pt-br/>

Logo Montadora	Relatório: 000001 Revisão: 0 Data: 06/05/2017 Página: 1 de 1		Logo Empresa Responsável									
	Projeto: Nome do Projeto	Cliente: Setores Clientes										
	Montadora: Nome Montadora	Contrato: Número do Contrato										
	LÍQUIDO PENETRANTE											
Plano de Inspeção: Número do Plano de Inspeção		Procedimento/Revisão: Número do Procedimento										
Norma de Ref.: Nome da Norma de Referência												
Crit. de Aceit.: Critério de Aceitação para a Norma		Cond. da Sup.: ESCOVADA										
Removedor: SOLVENTE		Método: TIPO 2 - VISIVEL										
Temperatura: 10 à 52°C		Iluminação: >1000										
Observação:												
IDENTIFICAÇÃO												
Unidade	Área	Isométrico	Spool	Junta	Rev.	Linha	L. Raiz	L. Acab.	M. Base	R.I.	Data	LPRem
NDPJ1	ROTA1	PC-F16-7038	SP05	019	0	PC-F16-7038	-	A	DP	-	06/05/2017	1S
NDPJ1	ROTA1	PC-F16-7038	SP05	019A	0	PC-F16-7038	-	A	DP	-	06/05/2017	1S
NDPJ1	ROTA1	PC-F16-7077	SP06	019	0	PC-F16-7077	-	A	DP	-	06/05/2017	1S
NDPJ1	ROTA1	PC-F16-7078	SP08	022	0	PC-F16-7078	-	A	DP	-	06/05/2017	1S
NDPJ1	ROTA1	PC-F16-7082	SP05	017	0	PC-F16-7082	-	A	DP	-	06/05/2017	1S

INSPETOR	COORDENADOR DO CQ	CLIENTE
Nome: Nome do Inspetor Certificação: Tipo de Certificação Código: Código de Referência	CARIMBO MONTADORA COORDENADOR DA QUALIDADE	

Figura 4. Exemplo de relatório de inspeção de líquido penetrante. Os campos textuais identificados são extraídos pela ferramenta proposta e mapeados para o banco de dados relacional.

como exclusivo, são inseridos apenas no primeiro envio do conjunto dos documentos referente a um projeto. A *relatorioinspecao* tem dados adicionados a cada relatório presente no documento (em geral, a cada duas páginas), sendo um elemento único o relatório que possui o mesmo grupo de identificação do documento e número a ele atribuído. As tabelas restantes abrangem uma lista de dados, que se conectam em uma sequência de linha, spool e junta. Sendo que as tabelas intermediárias *linha_spool* e *spool_junta* recebem as identificações que remetem às respectivas tabelas presentes no relacionamento. Em relação às tabelas que ligam *relatorioinspecao* com *spool_relatorio_inspecao* e *junta_relatorio_inspecao*, apenas uma delas será preenchida para cada relatório, pois o laudo, gerado pelo inspetor, aponta para o spool ou para a junta, dependendo do relatório. A tabela *carimbo*, a qual contém o conjunto nome e matrícula do inspetor como chave restrita, engloba uma lista de dados dos responsáveis pela obra.

5. CONCLUSÃO

Neste artigo foi proposto o desenvolvimento de uma função de mapeamento para representar em banco de dados relacional um conjunto de documentos digitalizados como imagem que representam relatórios de inspeção de construção naval.

O levantamento de requisitos gerou um modelo de dados generalista, que instruiu o desenvolvimento do mapeamento através da função de tradução. A utilização de

expressões regulares auxiliou na resolução de erros léxicos, retornados da extração dos dados pelo OCR.

O serializador elaborado permitiu a inserção, organizadamente e verificada, dos dados extraídos de um documento de inspeção, em um banco de dados relacional. Este processo, facilitou a busca por informações e suas relações para os demais setores. Também ajudou na análise de campos que os inspetores ou coordenadores ainda não preencheram, ou que não estão com os dados devidamente corretos.

As tecnologias utilizadas foram determinantes para implementação do protótipo como um conjunto de micro-serviços Web, os quais são facilmente executados num ambiente de nuvem computacional, otimizando recursos computacionais.

REFERÊNCIAS

- Brasil (1997). Lei nº 9.537, de 11 de dezembro de 1997. *Diário Oficial [da] República Federativa do Brasil*. URL https://www.planalto.gov.br/ccivil_03/leis/19537.htm.
- Costa e Silva, C. (2017). Métodos de inspeção de tubulações de transporte de minério: O caso da integridade de dutos. URL <http://www.monografias.ufop.br/handle/35400000/474>. Trabalho de Conclusão de Curso de Graduação, Universidade Federal de Ouro Preto, Ouro Preto - MG.

```
1 {
2   "projeto": {
3     "nome": "Nome do Projeto",
4     "data": "08/05/2017",
5     "contrato": "Numero do Contrato"
6   },
7   "relatorioinspecao": {
8     "data": "06/05/2017",
9     "numerorelatorio": "0000001",
10    "normaavaliacao": "Critério de Aceitacao"
11  },
12  "paginicial": 1,
13  "pagfinal": 1,
14  "idarquivo": "0000000abc",
15  "carimbo": true,
16  "assinatura": true
17 },
18 "linha": [
19   {
20     "numero": "PC-F16-7038"
21   }
22 ],
23 "spool": [
24   {
25     "numero": "SP05",
26     "numeroisometrico": "PC-F16-7038"
27   }
28 ],
29 "junta": [
30   {
31     "numero": "019"
32   }
33 ],
34 "linha_spool": [
35   {
36     "idlinha": "1",
37     "idspool": "1"
38   }
39 ],
40 "spool_junta": [
41   {
42     "idspool": "1",
43     "idjunta": "1"
44   }
45 ],
46 "spool_relatorio_inspecao": [
47   {
48     "laudo": "A",
49     "data laudo": "06/05/2017",
50     "idspool": "",
51     "idrelatorio": "1"
52   }
53 ],
54 "junta_relatorio_inspecao": [],
55 "carimbo": [
56   {
57     "nome": "Nome do Inspetor",
58     "matricula": "Codigo Inspetor",
59     "confiancaInspetor": 0.8
60   }
61 ]
62 }
```

Figura 5. JSON representando a inserção no banco de dados

Da, Zhongyue, Yang, Wenjie, Ran, Peipei, and Huo, Yuke (2017). Program design of json to structured data conversion. *MATEC Web Conf.*, 139, 00166. doi:

10.1051/mateconf/201713900166. URL <https://doi.org/10.1051/mateconf/201713900166>.

Dähling, S., Razik, L., and Monti, A. (2020). Owl2go: Auto-generation of go data models for owl ontologies with integrated serialization and deserialization functionality. *SoftwareX*, 12, 100571. doi:<https://doi.org/10.1016/j.softx.2020.100571>.

Elmasri, Ramez and Navathe, Shamkant B. (2018). *Sistemas de banco de dados*, volume 7 ed. Editora Pearson.

Geremia, G. (2012). Sistema autônomo de inspeção de dutos. URL <http://hdl.handle.net/10183/72898>. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, Porto Alegre - RS.

Kellogg, G., Champin, P.A., and Longley, D. (2019). *Jsonld 1.1-a json-based serialization for linked data*. Ph.D. thesis, W3C.

Roque, J.C.d.L. (2020). Avaliação do desempenho de técnicas não destrutivas: um estudo de caso na inspeção de juntas soldadas em tubulação de transferência de amônia anidra.

Silva, R., Suyama, F., Silva, R., and Delgado, M. (2020). Filtragem de dados de treinamento para aprendizagem profunda aplicada à inspeção de juntas soldadas em tubulações de petróleo. In *Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional*, 686–697. SBC, Porto Alegre, RS, Brasil. doi:10.5753/eniac.2020.12170. URL <https://sol.sbc.org.br/index.php/eniac/article/view/12170>.