

# Ferramenta de Pré-Processamento e Visualização de dados do DATASUS

Diego P. Mendes<sup>1</sup>, Fábio M. F. Lobato<sup>1,2</sup>, Antonio F. L. Jacob Jr.<sup>1</sup>

<sup>1</sup>Programa de Pós-Graduação em Engenharia da Computação (PECS) - Universidade Estadual do Maranhão (UEMA) – São Luís/MA

<sup>2</sup>Instituto de Engenharia e Geociências (IEG) - Universidade Federal do Oeste do Pará - Santarém/PA

**Abstract.** *DATASUS (Information Technology Department of the SUS) provides information that supports objective analyzes of the Brazilian health situation, assisting in evidence-based decision making and elaboration of health action solutions. In order to access this information, it is necessary to use the tools provided by the DATASUS portal that are very limited technically and complex. In the DATASUS portal, this information is in .DBC files and encrypted by means of a private algorithm, making it difficult to interpret the information. This work presents a tool developed in JAVA with the purpose of treating the public data provided by DATASUS and exporting them to a relational database in a simpler way. To this end, a tool was developed responsible for capturing, decrypting and exporting the information to a relational database.*

**Resumo.** *O DATASUS (Departamento de Informática do SUS) fornece informações que apoiam análises objetivas da situação sanitária brasileira, auxiliando em tomadas de decisões baseadas em evidências e elaboração de soluções de ações de saúde. Para acessar tais informações é necessário a utilização das ferramentas disponibilizadas pelo portal do DATASUS que são muito limitadas tecnicamente e complexas. No portal do DATASUS, essas informações estão em arquivos .DBC e criptografadas por meio de um algoritmo privado, dificultando a interpretação das informações. Este trabalho apresenta uma ferramenta desenvolvida em JAVA com o propósito de tratar os dados públicos disponibilizados pelo DATASUS e exportá-los para um banco de dados relacional de uma forma mais simples. Para este fim, foi desenvolvida uma ferramenta responsável pela captura, descryptografia e exportação das informações para um banco de dados relacional.*

## 1. Introdução

No Brasil, conforme Art. 5º, inciso XXXIII da Constituição Federal do Brasil, segundo [Federal 1988], “todos têm o direito de receber informações das esferas públicas para o seu uso particular e/ou coletivo, observando-se a legislação vigente e o caráter sigiloso das informações em relação à segurança da sociedade e do Estado brasileiro”.

Deste modo, conforme [Digital 2018], as ações do programa de Governo Eletrônico (eGOV) priorizam o uso das tecnologias da informação e comunicação (TICs)

para democratizar o acesso à informação, visando ampliar o debate e a participação popular na construção das políticas públicas, como também aprimorar a qualidade dos serviços e informações públicas prestadas.

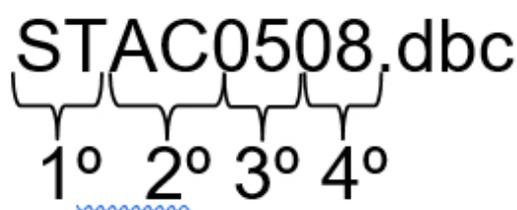
A política de Governo Eletrônico do Estado brasileiro segue um conjunto de diretrizes baseado em três ideias fundamentais: participação cidadã; melhoria do gerenciamento interno do Estado; e integração com parceiros e fornecedores.

Transformar a relação do governo com a sociedade e promover interatividade com cidadãos, empresas e órgãos governamentais melhora o processo de democratização do país, dinamiza os serviços públicos e proporciona uma administração pública mais eficiente, já que agora, a sociedade possui instrumentos para se manifestar junto às ações governamentais.

Segundo [DATASUS 2018], o Departamento de Informática do SUS (Sistema Único de Saúde) concede informações que podem contribuir em estudos e análises objetivas da situação da saúde brasileira, resoluções de problemas baseados em evidências e elaboração de programas de ações de saúde, dados de saúde (indicadores de saúde, informações epidemiológicas e de morbidade, assistência à saúde, estatísticas vitais, informações sobre a rede de assistência à saúde, informações demográficas e socioeconômicas) além de dados financeiros (referentes aos recursos do Fundo Nacional de Saúde transmitidos aos municípios, aos prestadores de serviços de saúde, aos orçamentos públicos de saúde declarados pelos Estados, pelo Distrito Federal e pelos Municípios). Essas bases de dados podem ser consultadas no portal do DATASUS.

A Tabela 1 informa todos os arquivos “.DBC” disponíveis no portal DATASUS.

Os arquivos disponibilizados no portal do DATASUS possuem um padrão em sua nomenclatura, uma forma de identificação de algumas informações através da sua descrição. Afim de facilitar a explicação, adotou-se como exemplo o arquivo exposto na Figura 1, com a divisão em blocos.



O diagrama mostra o nome de arquivo "STAC0508.dbc" dividido em quatro blocos por colchetes. Abaixo de cada bloco há um número ordinal: "1º" sob "STAC", "2º" sob "05", "3º" sob "08" e "4º" sob ".dbc". Uma linha ondulada azul está localizada sob o "2º".

Figura 1. Nomenclatura arquivo “.DBC”

- **Bloco 1:** subgrupo de arquivo;
- **Bloco 2:** UF do Estado;
- **Bloco 3:** mês da consulta;
- **Bloco 4:** ano da consulta.

No entanto os dados disponibilizados nem sempre estão em um arquivo que possa ser lido ou interpretável “humanamente”, necessitando assim de pré-processamentos para possibilitar a extração de informações, além de ser desconfortante a forma como DATASUS disponibiliza essas informações da saúde pública brasileira. Grande parte desses dados disponíveis gratuitamente estão no formato “.DBC”, uma versão do formato “.DBF”

<b>BASE</b>	<b>DESCRIÇÃO</b>
SIHSUS	Arquivos dissemináveis para tabulação do Sistema de informações Hospitalares do SUS
SIASUS	Arquivos dissemináveis para tabulação do Sistema de Informações Ambulatoriais do SUS
SIM	Arquivos dissemináveis para tabulação do Sistema de informações de Mortalidade
CIH	Arquivos dissemináveis para tabulação do Sistema de Comunicação de Informação Hospitalar
CIHA	Arquivos dissemináveis para tabulação do Sistema de Comunicação de Informação Hospitalar e Ambulatorial
SINASC	Arquivos dissemináveis para tabulação do Sistema de informação de Nascidos Vivos
SISPRENATAL	Arquivos dissemináveis para tabulação do Sistema de Monitoramento e Avaliação do Pré-Natal, Parto, Puerpério e Criança
CNES	Arquivos dissemináveis para tabulação do Cadastro Nacional de Estabelecimentos de Saúde

**Tabela 1. Dados disponíveis no portal DATASU**

comprimido, que dependem da utilização da ferramenta TABWIN para análise ou conversão para outros formatos, e uma ferramenta de linha de comando chamada DBC2DBF, limitadas ao sistema operacional WINDOWS, execução em máquina local do usuário, treinamento específico e o fato de não ser open source, impossibilitando personalizações de análises/consultas, implementação de melhorias, customizações no software e na escalabilidade das plataformas de software.

Analisando tais dificuldades presentes neste cenário, será apresentado uma nova ferramenta para facilitar/simplificar a captura, interpretação e análises dos dados do DATASUS sendo uma solução open source, multiplataforma (sem limitação de sistemas operacionais) e com uma apresentação mais amigável dos dados, facilitando assim sua interpretação e análise para tomadas de decisões.

## **2. Trabalhos Correlatados**

No Workshop de Transparência em Sistemas do ano de 2018, foi apresentado um trabalho com a finalidade de integrar e analisar as despesas do Governo Federal, segundo [Cristian Weiland 2018], quando entrou em vigor a Lei de Acesso a informação houve um crescimento na disponibilização de dados, porém juntamente com isso houve um aumento na dificuldade de tratar e interpretar tais informações. O trabalho apresenta o desenvolvimento do software SIMTransparencia, com o objetivo de fornecer uma nova opção de visualização, além do já existente Portal da Transparência do Governo Federal, exibindo dados de diversas bases em uma interface intuitiva e dinâmica.

A ferramenta apresentada neste trabalho deseja resolver a mesma problemática apresentada por [Cristian Weiland 2018], no que diz respeito a necessidade de buscar dados públicos, tratá-los e interpretá-los, porém quando se trata de dados públicos relacionados a saúde existe uma dificuldade a mais, pois para realizar o tratamento e a interpretação, é necessário a descryptografia dos dados, pois estes estão criptografados com um algoritmo privado, do qual se tem pouco conhecimento.

### 3. Metodologia

A metodologia empregada nesta ferramenta teve início com um levantamento de bibliografia e/ou trabalhos relacionados com a problemática de capturar, descryptografar os arquivos .DBC disponíveis no portal DATASUS, torná-los “entendíveis” humanamente e exportá-los para um banco de dados relacional.

O trabalho desenvolvido por [Adler 2017], foi a porta de entrada para o desenvolvimento de projetos que trabalham com a temática de leitura e interpretação de dados .DBC. Arquivos em formato .DBC são arquivos .DBF comprimidos e criptografados em algoritmos privados (PKWare), por exemplo os arquivos de dados DATASUS. Seu trabalho contribuiu com o desenvolvimento de um descompactador de arquivos .DBC que são condensados pela Biblioteca de Compressão PKWare, sendo até então o único algoritmo (escrito em C) capaz de descryptografar tais arquivos.

Com o algoritmo de descryptografia escrito em C do trabalho de Adler foi possível elaborar o desenvolvimento de uma ferramenta em linguagem JAVA. A linguagem JAVA oferece suporte nativo para a importação de funções e bibliotecas escritas em outras linguagens, como o C, C++ e FORTRAN. Aproveitando-se desta possibilidade, o código de descompressão de arquivos .DBC escrito em C foi transformado em uma dynamic link library (.DLL) para permitir o seu acesso pelo JNA *Java Native Access*), segundo [Dobrovkine 2018] é uma biblioteca que abstrai chamadas de métodos e os tipos de dados de uma linguagem para outra.

Na sequência, com o arquivo .DBC descryptografado e transformado em .DBF, utilizou-se a biblioteca “javadb-1.7.1” fornecida e mantida por [Fernández 2018] e outros desenvolvedores da comunidade, para realizar leitura e escrita de um arquivo “.DBF”, salientando-se que na biblioteca encontra-se uma classe DBFReader, que possui vários métodos para leitura de um arquivo “.DBF”, como o “getField” para recuperar os cabeçalhos do arquivo “.DBF” e o “nextRecord” para recuperar cada linha de dados do arquivo “.DBF”. O package java.sql apresentado por [ORACLE 2018], fornece a API (Application Programming Interface) para acessar e processar dados armazenados em uma fonte de dados (geralmente um banco de dados relacional) usando a linguagem de programação JAVA, a utilização dessa API facilitou a inserção dos dados lidos do arquivo .DBF em um banco de dados relacional.

A ferramenta desenvolvida neste trabalho foi validada e aprovada através de um método de averiguação do nível de usabilidade de um sistema, SUS (*System Usability Scale*), que segundo [USABILITY.GOV 2018], fornece uma ferramenta confiável e rápida para medir a usabilidade.

## 4. Arquitetura

Conforme a Figura 2, existem 2 etapas envolvidas para o funcionamento do sistema, que serão detalhadas posteriormente:

- **A importação dos dados (*datasusImport*):** onde todo o processo de download dos arquivos .DBC, sua descritografia e inserção em um banco de dados relacional será realizada;
- **A visualização dos dados (*datasusViewer* e *datasusViewer\_api*):** na qual é realizada toda interação com o usuário, que será capturada pelo *datasusViewer*, enviadas para o *datasusViewer\_api* e retornadas para o *datasusViewer*, possibilitando assim: criação de gráficos dinâmicos e exportação de planilhas, foi desenvolvida utilizando a tecnologia client-side, onde o lado do cliente (navegador) apenas dispara requisições HTTP (*Hypertext Transfer Protocol*) para uma API (*Application Programming Interface*), esta por sua vez retorna ou executa uma ação e o lado do cliente em posse dessas informações JSON (*JavaScript Object Notation*), rederizam a visão da página no navegador para o cliente.

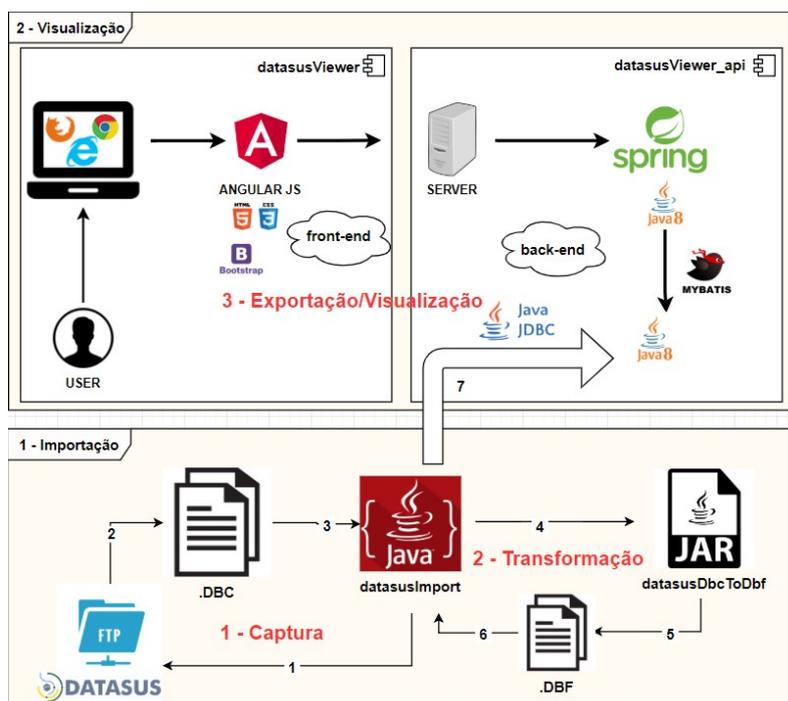


Figura 2. Arquitetura da ferramenta

A etapa da importação (projeto *datasusImport*) está dividida em 2 fases: a captura e a transformação dos dados .DBC, a etapa da visualização (projeto *datasusViewer*) em apenas 1 fase: a exportação.

Na fase da Captura é iniciado todo o processo, a ferramenta criada realiza o download das informações (.DBC) disponibilizadas em um endereço FTP (File Transfer Protocol) no portal do DATASUS e salva em uma pasta temporária, com a utilização da biblioteca *java.net*, que conforme [ORACLE 2018], fornece as classes para implementação de aplicativos de rede.

A seguir o detalhamento de cada campo, mostrado na Figura 3:

© 2018 DATASUS/CNES - Sistema de Relatórios. Todos os direitos reservados

**Figura 3. Tela de importação/transformação**

- **Mês:** campo onde deve ser informado qual mês deseja-se obter a informação. Ex: '01' = JANEIRO, '02' = FEVEREIRO;
- **Ano:** campo onde deve ser informado qual ano deseja-se obter a informação. Ex: '18' = 2018, '17' = 2017;
- **Estado:** campo onde deve ser informado qual estado deseja-se obter a informação. Ex: 'MA' = Maranhão, 'PI' = PIAUÍ;
- **Arquivo:** campo onde deve ser informado qual arquivo deseja-se obter a informação, no caso em questão estão sendo trabalhados os arquivos do domínio CNES. Ex: 'PF' = Profissionais, 'ST' – Estabelecimentos.

A fase da Transformação é a parte principal de operação da ferramenta. Após os arquivos .DBC serem “baixados” e salvos em uma pasta local inicia-se sua descritografia e posteriormente sua transformação. A extensão “.DBC” é uma compreensão dos arquivos “.DBF”. Esta compressão é realizada por meio de um algoritmo de compressão privado. No caso do DATASUS, é utilizado a biblioteca de compressão PKWare Compression. Neste sentido, utilizou-se o código “blast.c” e o “blast.h” desenvolvido na linguagem C por [Adler 2017], a fim de descritografar os arquivos .DBC.

Para que fosse possível a utilização deste algoritmo de descritografia escrita em C na estrutura da ferramenta desenvolvida em JAVA, utilizou-se o JNA (*Java Native API*), [Dobrovkine 2018], o qual permite um fácil acesso às bibliotecas nativas compartilhadas. O fim desta etapa resulta em um arquivo “.DBC” descritografado e transformado em um arquivo “.DBF”, podendo agora ser interpretado.

Para viabilizar o uso do código “blast”, primeiramente foi necessário a criação de uma DLL. Para criação da blast.c.dll foi utilizado o TDM-GCC 4.9.2 64-bit Release, conforme exposto na Figura 4. Após a criação da DLL (*Dynamic-Link Library*), a biblioteca passa a possuir uma classe principal chamada CBlastLibrary.java. Em sua construção, esta realiza o loadLibrary(“blast.c.dll”). Com isso, esta classe se beneficia da abstração

de toda complexidade da chamada de códigos nativos no Java. Devido esta facilidade pode-se acessar o código nativo do trabalho de (Adler, 2016) para descomprimir o .DBC e convertê-lo em .DBF.

```
$ gcc.exe -c blast.c -o blast.o -I "C:/Program Files (x86)/Dev-Cpp/MingGW64/include"
-I "C:/Program Files (x86)/Dev-Cpp/MingGW64/x86_64-w64-mingw32/include"
-I "C:/Program Files (x86)/Dev-Cpp/MingGW64/lib/gcc/x86_64-w64-mingw32/4.9.2/include" -DBUILDING_DLL=1

$ gcc.exe -c dbc2dbf.c -o dbc2dbf.o -I "C:/Program Files (x86)/Dev-Cpp/MingGW64/include"
-I "C:/Program Files (x86)/Dev-Cpp/MingGW64/x86_64-w64-mingw32/include"
-I "C:/Program Files (x86)/Dev-Cpp/MingGW64/lib/gcc/x86_64-w64-mingw32/4.9.2/include" -DBUILDING_DLL=1

$ gcc.exe -shared blast.o dbc2dbf.o -o dll.dll
-L "C:/Program Files (x86)/Dev-Cpp/MingGW64/include"
-L "C:/Program Files (x86)/Dev-Cpp/MingGW64/x86_64-w64-mingw32/include"
-static-libgcc -Wl, --output-def,libdll.def, --out-implib, libdll.a,-add-stdcall-alias
```

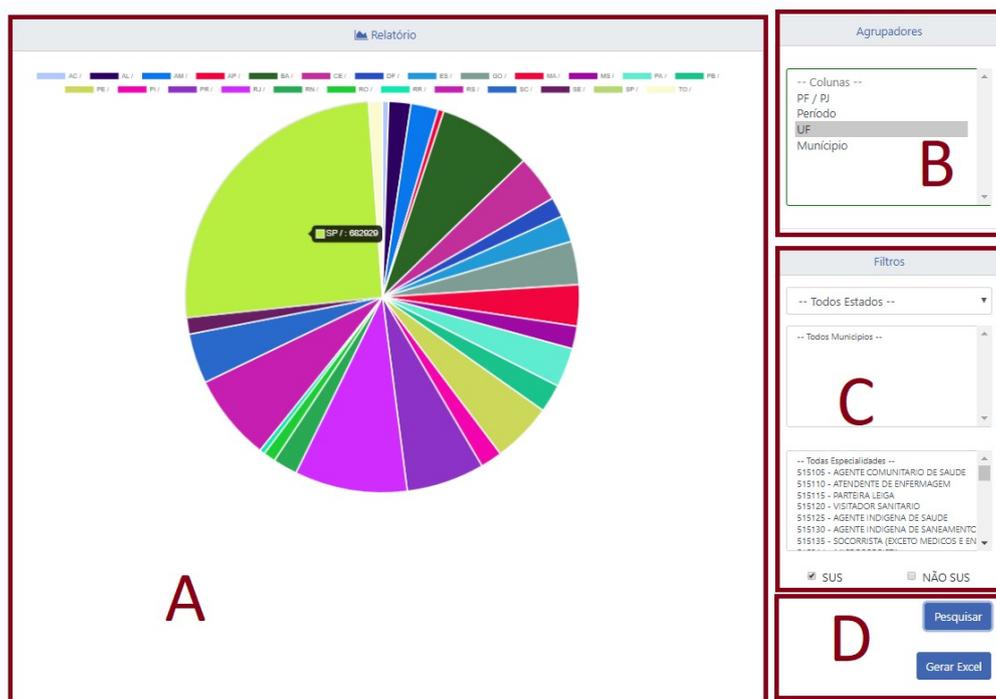
**Figura 4. Comandos criação blast\_c.dll**

A fase da exportação/visualização recebe como entrada o arquivo “.DBF” gerado na fase anterior e posteriormente cria uma tabela em um banco de dados com os cabeçalhos encontrados na primeira linha do arquivo, e assim, insere na tabela as informações encontradas nas linhas posteriores. As informações agora gravadas em um banco de dados ampliam a possibilidade de desenvolvimento de um software que permita a visualização de qualquer informação que seja interessante para subsidiar análises objetivas da situação sanitária, tomadas de decisão baseadas em evidências e elaboração de programas de ações de saúde.

Esta fase da arquitetura foi desenvolvida utilizando o padrão de arquitetura MVC (*Model – View - Controller*). Segundo [Gamma et al. 2000] o MVC é composto por três tipos de objetos. O modelo é o objeto de aplicação, a vista é a apresentação na tela e o controlador define a maneira como a interface do usuário reage às entradas do mesmo. Antes do MVC, os projetos de interface para o usuário tendiam em agrupar esses objetos. MVC para aumentar a flexibilidade e a reutilização. Para facilitar a construção da fase da visualização foram utilizados frameworks que agilizaram e facilitaram o desenvolvimento:

- **Angular JS:** é um framework de javascript mantido pela Google, que torna as telas do sistema mais dinâmicas, fazendo a renderização parcial da página, fazendo com que seja menor o consumo do tráfego na rede e otimizando o carregamento [Angular 2018];
- **JAVA:** é uma linguagem de programação multiplataforma compilada e interpretada por uma máquina virtual própria, com um vasto conjunto de bibliotecas de métodos e funções consolidados pela comunidade ativa da linguagem, possibilitando a realização de multitarefas [ORACLE 2018];
- **SPRING:** é um framework opensource para a plataforma JAVA. No Spring, o container se encarrega de ”instanciar” classes de uma aplicação Java e definir as dependências entre elas através de um arquivo de configuração em formato XML (*Extensible Markup Language*), inferências do framework, o que é chamado de *auto-wiring* ou ainda anotações nas classes, métodos e propriedades [SPRING 2018];
- **MYBATIS:** é uma estrutura de persistência de primeira classe com suporte a SQL customizado, procedimentos armazenados e mapeamentos avançados. O MyBatis elimina quase todo o código JDBC e configuração manual de parâmetros e recuperação de resultados [MYBATIS 2018].

Nesta etapa foram utilizados os arquivos “ST” (informações de estabelecimentos de saúde) e “PF” (informações de profissionais de saúde). A proposta desta aplicação visava demonstrar o panorama da distribuição de ofertas de especialidades médicas nos estados e municípios brasileiros utilizando somente os dados de 2018. A tela de relatórios é responsável pela montagem e visualização dessas informações em forma de gráfico ou exportação por meio de planilhas excel, a seguir alguns detalhes das possibilidades que podem ser aproveitadas.



**Figura 5. Tela de Relatórios**

O tela de relatórios se divide em quatro áreas:

- **Área A:** o resultado gráfico da consulta realizada, informando o quantitativo de agrupamento das colunas;
- **Área B:** local onde são escolhidos os agrupadores da consulta quantitativa, que serão as legendas do gráfico;
- **Área C:** filtros utilizados na consulta, que podem ser por estados, municípios, e especialidades médicas específicas que atendem pelo sistema SUS ou não.;
- **Área D:** ações possíveis da aplicação. A ação “Pesquisar” montará o resultado em gráfico e ação “Gerar Excel” exporta as informações em excel.

Para explanação de um dos objetivos, foi realizada uma consulta para analisar a distribuição de redes assistenciais de cardiologia pelo Brasil que atendem pelo sistema SUS. Foi notório observar que existe uma desproporcionalidade no que se diz respeito a distribuição da rede assistencial médica no Brasil, o caso analisado em questão mostrou que o estado de Roraima possui o menor quantitativo de disponibilidade de assistência cardiológica do Brasil, informação essa que fortalece a notícia divulgada segundo, [Brandão 2018], onde afirma que Roraima possui 1,56 médicos para cada mil habitantes. Por outro lado, São Paulo lidera com o maior número de assistência cardiológica

do país. Com essas informações em mãos pode-se explicar por exemplo o porquê da grande migração de pessoas para o estado de São Paulo em busca de soluções médicas.

Vale ressaltar que as ferramentas *datasusImport* - Sistema de Importação de dados do DATASUS para um banco de dados relacional e *datasusViewer* - Sistema de Visualização de dados do DATASUS advindos de um banco de dados relacional foram registradas juntos ao INPI (Instituto Nacional de Propriedade Intelectual).

## 5. Conclusão

O DATASUS possui uma enorme base de dados com os mais variados tipos de informações.

Apesar de serem públicos, a captura, interpretação e exploração desses dados não é tão trivial quanto parece, necessitando de utilização de ferramentas disponibilizadas pelo DATASUS, as quais possuem vários obstáculos e limitações técnicas, por exemplo: só podem ser executadas em ambiente WINDOWS, não são *opensource*, portanto não podem ser adaptadas de acordo com cada necessidade de quem usa e consultas muito limitadas.

A ferramenta proposta neste trabalho visa diminuir esses obstáculos e limitações, oferecendo a democratização dessas informações de uma forma mais clara e simples. Por ser *opensource* se apresenta como uma ótima opção para ser aperfeiçoada e até adaptada para interesses semelhantes ao proposto neste trabalho, podendo ser executada em qualquer sistema operacional e com o desempenho ainda maior.

Uma das suas maiores qualidades está na velocidade de captura,criptografia e armazenamento dessas informações em um banco de dados relacional ou planilhas, permitindo agora estudos na área da saúde que antes eram difíceis ou até impossíveis de serem realizados.

Para a avaliação da usabilidade da solução foi aplicado um questionário em 11 potenciais usuários da ferramenta (Cientista de dados e pesquisadores). Foi utilizado o *System Usability Scale (SUS)*, uma das mais conhecidas e simples metodologia de averiguação do nível de usabilidade de um sistema.

Conforme [USABILITY.GOV 2018] a média de score é 68 pontos. Se o score obtido estiver abaixo dessa média, provavelmente o software está enfrentando problemas sérios de usabilidade. A ferramenta alcançou um bom resultado (score de 77,5), demonstrando que a ferramenta é efetiva, eficiente e satisfatória, segundo a escala SUS.

## Referências

- Adler, M. (2017). A massively spiffy yet delicately unobtrusive compression library. Disponível em: <<https://github.com/madler/zlib/tree/master/contrib/blast>>. Acesso em: 01 fev. 2018.
- Angular (2018). Angular - one framework mobile and desktop. Disponível em: <<https://angular.io>>. Acesso em: 14 dez. 2018.
- Brandão, I. (2018). Número de médicos em roraima é 29% menor que a média nacional, diz crm-rr — roraima — g1. Disponível em: <<https://g1.globo.com/rr/roraima/noticia/>>

numero-de-medicos-em-roraima-e-29-menor-que-a-media\  
\-nacional-diz-crm-rr.ghtml>. Acesso em: 15 nov. 2018.

Cristian Weiland (2018). Integração e Análise das Despesas do Governo Federal. *Anais do Workshop de Transparência em Sistemas (WTRANS)*.

DATASUS (2018). Cnes - datasus. Disponível em: <<http://datasus.saude.gov.br/sistemas-e-aplicativos/cadastrros-nacionais/cnes>>. Acesso em 01 mar. 2018.

Digital, G. (2018). O conceito de governo eletrônico. Disponível em: <<https://www.governodigital.gov.br/EGD/historico-1/historico>>. Acesso em: 01 fev. de 2018.

Dobrovkine, D. (2018). Java native access. Disponível em: <<https://github.com/java-native-access/jna>>. Acesso em: 01 nov. 2018.

Federal, S. (1988). Constituição da república federativa do brasil. Disponível em: <[https://www.senado.leg.br/atividade/const/con1988/CON1988\\_05.10.1988/CON1988.pdf](https://www.senado.leg.br/atividade/const/con1988/CON1988_05.10.1988/CON1988.pdf)>. Acesso em 19 mar. 2018.

Fernández, A. (2018). Java library for reading and writing xbase (dbase/dbf) files. Disponível em: <<https://github.com/albfernandez/javadbff>>. Acesso em: 28 dez. 2017.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (2000). *Padrões de Projeto - Soluções Reutilizáveis de Software Orientado a Objetos*. 1th edition.

MYBATIS (2018). What is mybatis? Disponível em: <<http://www.mybatis.org/mybatis-3>>. Acesso em: 15 dez. 2018.

ORACLE (2018). Java native interface specification-contents. Disponível em: <<https://docs.oracle.com/javase/7/docs/technotes/guides/jni/spec/jniTOC.html>>. Acesso em: 15 dez. 2018.

SPRING (2018). Spring: the source for modern java. Disponível em: <<https://spring.io>>. Acesso em: 15 dez. 2018.

USABILITY.GOV (2018). System usability scale (sus). Disponível em: <<https://spring.io>>. Acesso em: 6 jan. 2018.