

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO  
CÂMPUS CAMPINAS

Arthur Pereira Rozado

**Implementação de infraestrutura de API para extração de dados tabulares a partir de  
documentos PDF**

CAMPINAS

2017

Arthur Pereira Rozado

**Implementação de infraestrutura de API para extração de dados tabulares a partir de documentos PDF**

Trabalho de Conclusão de Curso apresentado como exigência parcial para obtenção do diploma do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia Câmpus Campinas.

Orientador: Andreiuid Sheffer Corrêa

Coorientador: Henrique Gonçalves Salvador

CAMPINAS

2017

R893i Rozado, Arthur Pereira  
Implementação de infraestrutura de API para extração de dados tabulares a partir de documentos PDF / Arthur Pereira Rozado. – Campinas, 2017. 38f. : il.

Orientador: Andreiuid Sheffer Corrêa.  
Coorientador: Henrique Gonçalves Salvador.

Monografia (Graduação) – Instituto Federal de São Paulo – Câmpus Campinas. Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, 2017.

1. PDF – Formato de arquivo. 2. Interface de programas aplicativos. 3. CSV – Formato de documento. 4. Dados abertos. I. Instituto Federal de São Paulo - Câmpus Campinas. Curso de Tecnologia em Análise e Desenvolvimento de Sistemas. II. Título.

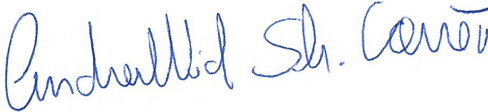
Arthur Pereira Rozado

**Implementação de infraestrutura de API para extração de dados tabulares a partir de documentos PDF**

Trabalho de Conclusão de Curso apresentado como exigência parcial para obtenção do diploma do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia Câmpus Campinas.

Aprovado pela banca examinadora em: 15 de dezembro de 2017

**BANCA EXAMINADORA**



---

Prof. Dr. Andreiuid Sheffer Corrêa (orientador)  
IFSP Câmpus Campinas

Prof. Dr. Tiago José de Carvalho  
IFSP Câmpus Campinas

Prof. Dr. Ralph Santos da Silva  
Centro da Tecnologia da Informação Renato Archer

*“Dedico este trabalho para a minha família e todos os que participaram da minha vida nos últimos três anos”.*

## **AGRADECIMENTOS**

Agradeço em primeiro lugar ao meu orientador Prof. Dr. Andreiuid Sheffer Corrêa e ao meu coorientador Prof. Me. Henrique Gonçalves Salvador, que me orientaram e apoiaram durante o desenvolvimento do trabalho.

Gostaria de agradecer ao Instituto Federal São Paulo por fornecerem tudo o que me foi necessário para alcançar meus objetivos. Agradeço também a todos os professores que me auxiliaram na minha formação e todos os que participaram do meu desenvolvimento pessoal e do desenvolvimento do projeto.

E por último, mas não menos importante, agradeço aos meus pais e minha família que me serviram de base e exemplo e me apoiaram, permitindo que eu continuasse meus estudos.

“A mente que se abre a uma nova ideia jamais voltará ao seu tamanho original”.

Albert Einstein

## RESUMO

O movimento de dados abertos vem sendo consolidado nos últimos anos para definir requisitos para promover uso, a reutilização e redistribuição dos dados por qualquer um e para qualquer propósito. Em vários países de todo o mundo, a área governamental vem tomando frente neste movimento com iniciativas de divulgar informações de transparência atendendo aos requisitos de dados abertos com apoio das legislações específicas. O problema é que o atendimento dos requisitos de dados abertos é algo que demanda tempo e preparação dos agentes públicos. Com isso, tem-se informações sendo divulgadas, e não dados, o que compromete os benefícios pretendidos. Um dos principais formatos preferidos é o Portable Document Format (PDF), indicado somente para leitura humana. Este trabalho objetiva implementar uma infraestrutura composta de Application Programming Interfaces (APIs) para extração de dados tabulares e convertê-los em formatos compatíveis com dados abertos. O sistema está limitado a algumas deficiências das bibliotecas utilizadas, porém, permite conversão e alimentação de uma base colaborativa por meio de múltiplas plataformas. Com os resultados deste trabalho, a comunidade poderá utilizar as interfaces disponibilizadas para utilização por outros sistemas sem limitações de linguagens e tecnologias.

**Palavras-chave:** Dados abertos. Tabula. CSV.



## **ABSTRACT**

The movement of open data has been consolidated in the last years to define requirements to promote of data's use, reuse and redistribution by any and for any purpose. The government area has taken up this move with initiatives to disseminate transparency information in response to open data requirements with the support of specific legislation. The problem is that meeting the requirements of open data is something that demands public agents' time and preparation. So we have information being disclosed, not data, which compromises the intended benefits. One of the main formats is the Portable Document Format (PDF), which is indicated only for human reading. This work aims to implement an infrastructure composed of Application Programming Interfaces (APIs) for extracting tabular data and converting them into formats compatible with open data. The system is limited to some shortcomings presents in the libraries that were used, but it allows conversion and feeding of a collaborative base through multiple platforms. With the results of this work, the community will be able to use the interfaces available for use by other systems without limitations of languages and technologies.

**Keywords:** Open data. Tabula. CSV.

## LISTA DE FIGURAS

Figura 1 – Exemplo de documento CSV. . . . .	19
Figura 2 – Estrutura dos serviços da API. . . . .	22
Figura 3 – Código de implementação de um dos controladores da API. . . . .	24
Figura 4 – Código de implementação da classe que será enviada como JSON. . . . .	24
Figura 5 – Resposta da API sobre requisição "version". . . . .	24
Figura 6 – Exemplo de resultado do uso da chamada “buscar/mes”. . . . .	28
Figura 7 – Exemplo de resultado do uso da chamada “arquivo”. . . . .	29
Figura 8 – Execução da chamada “-v” do Tabula que retorna a versão do mesmo. . . .	30
Figura 9 – Pagina inicial do CKAN . . . . .	31
Figura 10 – Tabela com células mescladas. . . . .	33
Figura 11 – Tabela com células mescladas após conversão pelo Tabula. . . . .	33

## **LISTA DE TABELAS**

Tabela 1 – Representação dos dados de um CSV. . . . .	19
Tabela 2 – Métodos do sistema . . . . .	25

## LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
CSV	<i>Comma-separated values</i>
HTML	<i>HyperText Markup Language</i>
PDF	<i>Portable Document Format</i>
URL	<i>Uniform Resource Locator</i>

## LISTA DE SÍMBOLOS

*mb*            megabytes

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>14</b>
1.1	OBJETIVOS . . . . .	15
1.1.1	Objetivo geral . . . . .	16
1.1.2	Objetivos específicos . . . . .	16
1.2	JUSTIFICATIVA . . . . .	16
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>18</b>
<b>3</b>	<b>MATERIAIS E MÉTODOS . . . . .</b>	<b>22</b>
3.1	MATERIAIS . . . . .	23
3.2	MÉTODOS . . . . .	23
3.2.1	api e json . . . . .	23
3.2.2	Tabula . . . . .	29
3.2.3	ckan . . . . .	30
<b>4</b>	<b>TESTES . . . . .</b>	<b>32</b>
<b>5</b>	<b>RESULTADOS . . . . .</b>	<b>33</b>
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>35</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>36</b>

## 1 INTRODUÇÃO

O movimento de dados abertos no Brasil representa uma área de grande importância no cenário da atualidade, onde graças ao portal de transparência governamental brasileiro, que visa garantir acesso ao conteúdo de propriedade pública de forma livre, temos acesso a dados governamentais de grande importância, como por exemplo planilhas de gastos mensais, semestrais e anuais de cidades e estados brasileiros. De acordo com o Painel de Monitoramento de Dados Abertos' (TRANSPARÊNCIA, 2017), está previsto até o final de 2017 a abertura de 2104 bases de dados públicos, das quais 76% já estão disponíveis para acesso em outubro de 2017.

De acordo com a Global Opendata Index (INTERNATIONAL, 2016), pesquisa realizada anualmente a partir de 2013 responsável por analisar e avaliar o sistema de dados abertos de países pela Open Knowledge Foundation, de 2015 para 2016 o Brasil subiu quatro posições entre os países analisados, passando a atuar como líder em compartilhamento de dados abertos da América Latina e ocupa a oitava posição na classificação mundial, compartilhando um total de 68% de todos os dados gerados pelo governo.

Para que a disponibilização de dados seja compatível com dados abertos, além de disponibilizar a massa de dados, deve-se seguir requisitos previamente definidos para que os dados (provenientes da transparência pública) possam ser livremente usados, reutilizados e redistribuídos, por qualquer um, para qualquer propósito (OPEN KNOWLEDGE FOUNDATION, 2012; TAUBERER, 2014).

O movimento de dados abertos propõe uma série de requisitos para guiar a abertura de registros públicos com o uso de infraestrutura específica de software. O Índice de Dados Abertos para o Brasil (Brasil, 2017) de 2017, documento criado pela Open Knowledge Brasil (Brasil, 2017), analisa o cenário brasileiro e expõe algumas deficiências presentes no sistema de dados abertos atual, onde a dificuldade de trabalhar os dados está presente em 9 das 12 áreas de dados abertos disponíveis. Além da literatura disponível, existe toda uma legislação própria para tratar do assunto, como exemplo a Lei de Acesso à Informação (Lei nº 12.527/2011) e o decreto Nº 8.777, de 11 de maio de 2016 em vigência no cenário brasileiro.

Apesar dos potenciais benefícios com a disponibilização de dados abertos, Coelho et al. (2015) revelam que a publicação de dados governamentais compatíveis com dados abertos no Brasil ainda é incipiente. Percebe-se a proliferação de websites de transparência na contramão dos

dados abertos. Estes websites mostram-se verdadeiros repositórios de documentos semelhantes a relatórios impressos, normalmente viabilizados através de Portable Document Format(PDF) e HyperText Markup Language(HTML) (Corrêa; Corrêa; Silva, 2014). Mesmo com o excelente posicionamento no ranking mundial, o Brasil ainda se depara com sérios problemas relacionados com a compatibilidade dos arquivos disponibilizados e portais de transparência que não seguem nenhum padrão, o que resulta em métodos que dificultam o uso e acesso aos dados fornecidos.

Para tratar o problema da impossibilidade de trabalhar e tratar os dados, devido ao formato fechado utilizado e incompatibilidade com leitura por máquina do mesmo, foi implementado uma infraestrutura composta de Application Programming Interfaces(APIs) que fornecem serviços de extração de dados tabulares contidos em documentos PDFs, que é um formato fechado, incompatível com dados abertos e é atualmente um dos principais formatos utilizados para disponibilização de documentos. A extração dos dados resulta em um documento no formato Comma-Separated Values(CSV), amplamente conhecido, universal e compatível com dados abertos. Considerando a simplicidade e compatibilidade do formato, acredita-se que a utilização do CSV é um passo para que os dados possam ser livremente usados e redistribuídos para qualquer propósito. Além da conversão do arquivo para formato aberto, o sistema gerencia o armazenamento e compartilhamento dos arquivos CSV por meio da plataforma CKAN de forma aberta e colaborativa, onde a base de dados é alimentada pela comunidade e interessados nas informações, que permanecem disponíveis para acesso livre.

No Capítulo 2 os materiais é abordado a estrutura do sistema, considerando formatos e tipos de dados utilizados. De forma superficial, tudo o que foi utilizado é abordados. O Capítulo 3 aborda as ferramentas e as formas como elas se comunicam por meio do desenvolvimento das camadas de API. Os testes são definidos no Capítulo 4, onde é ressaltado o uso de uma massa de dados utilizada e modos de testes. A conclusão pode ser encontrada no Cap. 6.

## 1.1 OBJETIVOS

Como objetivos principais, o sistema deve converter dados tabulares de documentos PDF para formato aberto e compatível com leitura por máquina, compartilhar de forma simples e acessível os dados e facilitar o uso em softwares.



### 1.1.1 Objetivo geral

Desenvolver um sistema que permita conversão e compartilhamento de dados tabulares abertos em formato compatível com o padrão de dados abertos.

### 1.1.2 Objetivos específicos

1. Selecionar materiais open-source para permitir que todo o código do programa seja aberto, possibilitando assim reuso.
2. Desenvolver infraestrutura para conversão de dados se baseando no método de camadas.
3. Converter dados tabulares com fidelidade ao documento original, sem perda de dados.
4. Disponibilizar um sistema colaborativo onde dados são fornecidos e consumidos pela comunidade, utilizando uma infraestrutura de API para permitir consumo dos dados amplamente em outros softwares e plataformas.
5. Permitir uso aberto e livre do sistema de APIs desenvolvido.

## 1.2 JUSTIFICATIVA

Considerando o cenário governamental atual do Brasil, onde se faz necessário um certo acompanhamento de dados públicos, uma grande massa de dados se torna disponível para acesso livre, permitindo assim ampla participação ativa do povo e outros interessados. A deficiência de formato de documento prioritário, fechado e incompatível com leitura por máquina, lesa os usuários do sistema de transparência e torna o uso dos dados em sistemas de informação inviável. Essa deficiência é tratada por padrões de dados abertos governamentais como crítico, considerando o potencial uso das informações fornecidas. A utilização de uma alternativa que torne possível tratar os dados, convertendo e adequando para formato compatível e aberto por meio computacional, se faz justificável não apenas pelo valor dos dados em questão, mas também pela quantidade de dados, que de acordo com o painel de monitoramento de dados abertos brasileiros (TRANSPARÊNCIA, 2017) chegam a atingir 2227 bases de dados para abertura até o final de 2017, onde atualmente 1656 já estão disponíveis para acesso.

A ampla base de dados disponibilizada, oferece uma taxa de 68% de compatibilidade com dados abertos. Considerada como uma das maiores deficiências dos portais de transparência, a incompatibilidade com leitura por máquina afeta diretamente todo o processo e compromete a

estrutura proposta, e por isso, um sistema capaz de corrigir ou auxiliar no processo de migração para a total compatibilidade com o proposto pelos dados abertos, pode ser de grande ajuda no caminho para alcançar o ideal quando o assunto é dados governamentais abertos e o próprio projeto de transparência. Não existe no Brasil, nenhum exemplo de portal de transparência compatível com os modelos propostos, ou seja, nenhum dos portais de transparência que disponibilize os dados integralmente adotando os formatos abertos.

Um sistema padronizado para compartilhamento dos dados também se torna uma necessidade, considerando que grande parte dos portais de compartilhamento de dados abertos governamentais oferecem dificuldades para acesso dos dados, além da impossibilidade de adquirir grandes quantidades de informações de uma única vez, tornando-se assim, mais uma deficiência no caminho dos dados abertos.

Desta forma, a implementação do sistema em camadas permite que o fornecimento de dados seja compatível com os portais governamentais e sistemas de outros desenvolvedores, permitindo também o uso via CKAN, o que torna justificável o sistema e a forma de implementação, considerando a resolução dos problemas de compatibilidade do formato do arquivo, resultando em arquivos que permitem o uso dos dados e em formato aberto, além de permitir padronização do método de compartilhamento dos dados e a possibilidade de trabalhar com dados em ampla escala em softwares considerando o uso da API desenvolvida.

## 2 FUNDAMENTAÇÃO TEÓRICA

Iniciando a discussão, existem duas leituras necessárias para contextualização com os temas dados abertos e dados abertos governamentais. De acordo com Bennett & Harvey (2009), o terceiro passo para compartilhamento de dados abertos governamentais é “faça dados legíveis por humanos e por máquinas”. De acordo com o OPEN DATA HANDBOOK(FOUNDATION, 2014) reutilizar dados do setor público não deve ser sujeito a restrições de nenhum tipo. Além disso, é de extrema importância que as informações sejam providas em formato legíveis por máquina, para permitir que a reutilização seja utilizada ao máximo. Como exemplo, dados publicados como documentos no formato PDF (Portable Document Format), um tipo de arquivo com dados que podem ser facilmente interpretados por seres humanos, mas oferecem dificuldades para uso computacional das informações. Desta forma, é imposto um limite com relação as possibilidades de reutilização dos dados.

Considerando o PDF como um formato incompatível com dados abertos, a OPEN KNOWLEDGE INTERNATIONAL, define o formato que será utilizado para transição de dados entre o servidor e usuário como compatível e acrescenta citando que o JSON é um formato de arquivos simples e de leitura simples, independente da linguagem de programação que é usada. De forma direta, sua simplicidade representa facilidade para que os dados sejam lidos por computadores, mais até do que outros formatos como o XML.

Ainda tratando os formatos de arquivo, o formato CSV é recomendado para compartilhamento de dados tabulares abertos, expondo não apenas a compatibilidade do formato, mas também a capacidade de um documento de representar de forma fiel e sucinta dados tabelares, para permitir a leitura sem danificar o conteúdo, o que torna o documento CSV totalmente compatível com o projeto. Por ser compacto se torna adequado para transportar uma grande quantidade de dados com estrutura semelhante. Porém o formato é tão bruto que os dados podem se tornar inutilizáveis sem uma documentação, considerando a dificuldade de identificar o significado dos dados de cada coluna. Graças a isso, existe uma necessidade de descrição dos dados para uso correto. Se torna essencial que a estrutura do arquivo seja respeitada, considerando que a omissão de um único campo pode perturbar o uso e leitura dados presentes no arquivo, pois não se pode definir a forma como interpretar os dados remanescentes.

Porém, mesmo com as possibilidades de uso do CSV, ele ainda apresenta alguns defeitos por ser um formato bruto de texto puro. É um formato de arquivo dedicado a softwares de

dados tabulares, utilizando texto puro e formatado usando virgula para separar células da tabela e quebra de linha no documento de texto para diferenciar as linhas da tabela. Um exemplo do formato CSV pode ser visto na Figura 1.

Figura 1 – Exemplo de documento CSV.

```
Vermelho, 100, "carros, flores",, verdadeiro
Azul, 200,,1.5, falso
,,,
Amarelo,,,verdadeiro
```

Fonte: Produzido pelo autor

A Tabela 1 é o resultado do CSV representado acima em um software para dados tabulares.

Tabela 1 – Representação dos dados de um CSV.

Vermelho	100	carros, flores		verdadeiro
Azul	200		1.5	falso
Amarelo				verdadeiro

Fonte:Desenvolvido pelo autor (2017)

Assim, é possível trabalhar normalmente com os dados realizando todas as operações matemáticas necessárias, aplicando formulas e até mover células, mesmo com tabelas muito maiores e com mais dados.

Além de fornecer informações que garantem a compatibilidade dos documentos encontrados, o OPEN DATA HANBOOK (FOUNDATION, 2014) descreve que um arquivo “fechado”, pode ser resultado de um formato proprietário e sua especificação e documentação pode permanecer indisponível publicamente, ou até permanecer disponível, mas seu reuso é limitado. Se os dados forem disponibilizados em um arquivo com formato fechado, pode-se obter como resultado significativos obstáculos ao reuso, criando a necessidade de compra do software aos que desejam usar esta informação.

Além disso, a necessidade da compatibilidade com formatos abertos é ressaltada na exaltação do benefício dos formatos abertos de arquivo, que permitem aos desenvolvedores produzir múltiplos serviços e softwares que utilizem estes dados de forma livre permitindo assim, minimizar os obstáculos ao reuso da informação contidas nos arquivos. A W3C (Bennett; Harvey, 2009) refere-se ao formato de documento aberto citando a necessidade da compatibilidade com máquina e eficiência do formato “[...]Quando possível, use padrões abertos estabelecidos e ferramentas que permitam uma produção e publicação fácil e eficiente dos dados.[...]” para permitir que os dados possam ser utilizado por todos.

Tratando-se dos dados abertos, MANOCHA (2011) diz que transparência é um bom começo, mas só é possível avaliar o valor real dos dados se forem devidamente explorados. Desta forma, torna-se essencial que os dados possam ser analisados para serem úteis, de forma a ser o próximo passo que o governo precisa tomar para a abertura e melhor entrega do serviço público.

Ainda, de acordo com Coelho et al. (2015), existem muitos potenciais benefícios na utilização dos dados abertos, porém a quantidade de dificuldades a se enfrentar para alcançar o ideal no quesito transparência ainda é grande. É possível notar que as TIC são essenciais para ampliar a transparência do Estado, notar um espaço existente para avanços, e esta observação é acentuada para os governos locais. Muito pode ser aprendido com os primeiros esforços dedicados a uma maior accountability e transparência.

A definição de accountability é dada como a “[...]obrigação de os funcionários públicos informarem sobre o uso dos recursos públicos e responsabilização do governo ao público[...]”. De acordo com Silva Ribeiro e Almeida (RIBEIRO, ), este é um esforço que permitirá que além do acesso aos dados, exista a possibilidade de o público geral compartilhar e utilizar os dados automaticamente, sem a necessidade de intervenção humana. Desta forma, existirá a possibilidade de gerar novos conhecimentos, novos produtos e serviços, viabilizando, inclusive, o aumento do papel da sociedade de fiscalização, realizada por meio de verificação e validação dos dados oficiais fornecidos.

Os autores Vaz, Ribeiro e Matheus, referem-se as TICs como uma porta favorecendo o sistema de transparência onde é possível notar que o desenvolvimento das TICs (tecnologias de comunicação e informação), nas últimas décadas, trazem novas possibilidades para o desenvolvimento e aplicação da transparência. Com auxílio das TICs, como a Internet, é possível potencializar a promoção da transparência, já que os meios eletrônicos resultam em uma maior facilidade de acesso aos dados e informações da Administração Pública.

Mesmo com todos os benefícios envolvendo transparência e dados abertos, ainda é possível notar grandes falhas no processo de abertura de dados, que afetam não apenas o usuário final, mas também comprometendo todo o processo e os objetivos do mesmo, como o mostrado pelo OPEN DATA INDEX (INTERNATIONAL, 2016), responsável por analisar e avaliar o desempenho de países na abertura dos dados. É possível notar uma grande deficiência no compartilhamento de dados em formato aberto e legível por máquina. A OPEN KNOWLEDGE BRASIL, cita no documento de relatórios anual o processo de transparência governamental como um processo em desenvolvimento e em constante evolução onde foram encaminhadas melhorias dos processos, canais de comunicação, participação e instâncias de decisão da organização, visando ampliar e favorecer a transparência, mantendo respeito pelos acordos com parceiros, prestadores de serviços e financiadores.

Porém, a deficiência permanece sem alterações, mesmo com a quantidade de dados aumentando constantemente. Afirmado assim mais uma vez a necessidade de dados úteis e utilizáveis, torna-se necessário a construção de um ambiente capaz e estrutura com capacidade de comportar tais informações. O desenvolvimento de tal estrutura, foi realizada se baseando nas diretrizes propostas por Corrêa, Corrêa e da Silva (2015), documento no qual é proposto pelos autores uma infraestrutura composta de APIs que permitam compatibilidade e padronização com o padrão de dados abertos convertendo dados em um formato fechado e legível apenas para humanos, em um formato aberto compatível com leitura para ambos, humano e máquina.

Tratando-se da API, existem muitas leituras disponíveis para tratar o assunto, de acordo com Brian Mulloy (2016)

“Como o REST é um estilo arquitetônico e não um padrão estrito, ele permite uma grande flexibilidade. Por causa dessa flexibilidade e liberdade de estrutura[...]” considerando a facilidade de uso fornecida para os desenvolvedores, e ampla compatibilidade com linguagens de programação, a REST se torna a arquitetura perfeita para compartilhar amplamente dados via software. Uma API é basicamente um sistema que fornece acesso a um serviço, sistema ou dado via web, dispensando assim a necessidade de o usuário compreender o que está sendo feito pelo sistema e mesmo assim utiliza-lo, necessitando apenas do conhecimento básico para usar o sistema.

Uma REST ou RESTful API é um formato de API que recebe requisições HTML e como resposta transmite dados em formato compatível com todas as linguagens de programação e plataformas. Independentemente de quem receba a informação enviada, ela pode ser assimilada e trabalhada facilmente, sem a necessidade de grandes conversões ou de conhecimento do sistema que fornece a informação. Desta forma é possível trabalhar com massas de dados em múltiplas formas, apenas consumindo o que é fornecido pela API. As requisições são realizadas usando URLs que acessam ao servidor requisitando o serviço da API, e como retorno são obtidos os dados. A imagem a seguir representa de forma simplificada a estrutura da API.

A API desenvolvida com Spring, permite tratar requisições com facilidade, realizar buscas entre os dados disponíveis e enviar esses dados em formato JSON, que é basicamente um documento de texto com uma estrutura específica que usa um sistema de chave-valor para comportar informações de qualquer tipo primitivo, ou seja números e palavras.

A forma escolhida para compartilhar os documentos convertidos com o usuário final foi o CKAN, um sistema capaz de gerenciar dados e torna-los acessíveis, fornecendo ferramentas para racionalizar publicação, compartilhamento, busca e uso das informações. E além disso, de acordo com o próprio site do CKAN, é um sistema com código aberto e livre. O que quer dizer que pode ser usado sem taxas de licença, e todos os direitos sobre os dados e os metadados inseridos são do autor ou desenvolvedor, garantindo assim transparência, padronização, facilidade e compatibilidade do sistema.

Carvalho (2015) define o CKAN como uma ferramenta de gestão de dados que objetiva tornar informação acessíveis para a comunidade por meio de ferramentas para publicação, compartilhamento, pesquisa e reutilização dos dados. Mantém um foco em fornecedores de dados de âmbitos variados, podendo envolver, organizações, empresas, consórcios regionais ou instituições de nível nacional que pretendam disponibilizar e partilhar os seus dados. Afirma também que o CKAN é um sistema de código aberto com grande capacidade de adaptação, escrito em Javascript e Python, e tem foco particular em portais de dados governamentais. Além do sistema de armazenamento e gerenciamento dos dados, o CKAN ainda conta com uma API própria, e com a disponibilização de interfaces para obtenção e disseminação de informação entre máquinas, a API do CKAN possibilita acesso à informação dos conjuntos de dados de forma a permitir a leitura, atualização ou adição de informações para utilizadores autorizados.

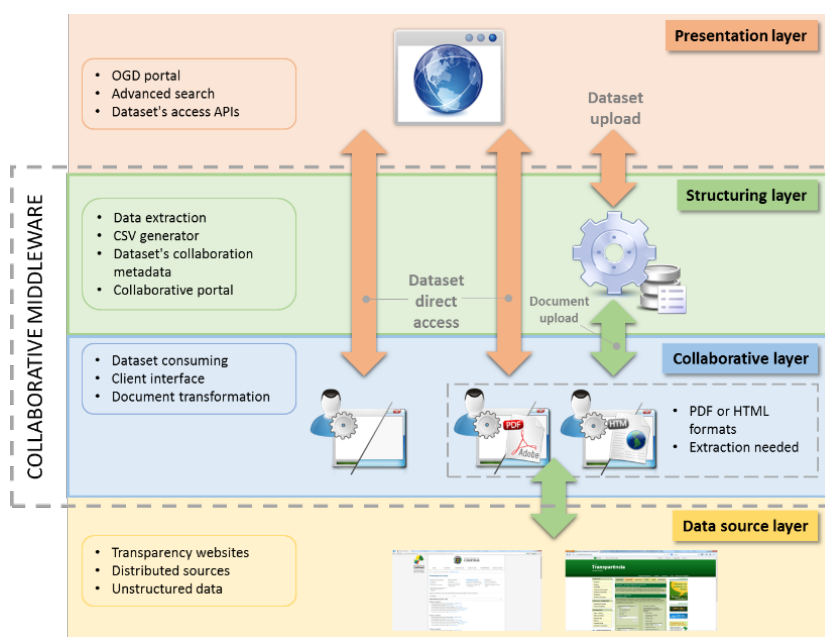
Sobre a questão legal, atualmente o Brasil conta com três leis e nove decretos, responsáveis por garantir o direito ao acesso livre e aberto aos dados governamentais. A lei mais atual é responsável por regular e garantir o acesso a informação.

### 3 MATERIAIS E MÉTODOS

De acordo com a infraestrutura proposta, o sistema de APIs trata diretamente uma grande massa de dados que é fornecida ao sistema pelos usuários de forma colaborativa. A estruturação do sistema pode ser notada na Figura 2, tendo como ênfase os serviços de extração dos dados, conversão para CSV e interação com a Camada de Apresentação. A implementação é baseada nas camadas de APIs, armazenamento de dados e gerenciamento dos mesmos, resultando em uma API que permita o desenvolvimento de uma camada de apresentação independente do sistema, que consuma os dados fornecidos pelos mesmos sem conhecimento amplo de seu funcionamento. A camada de apresentação deve ser implementada utilizando os padrões de linguagens para uso de API, e pode ser realizada em qualquer linguagem que permita o uso das mesmas.

A implementação do sistema em camadas é fundamentada na arquitetura proposta por Corrêa, Corrêa & Silva (2014), onde é possível notar a modularização das camadas do sistema, permitindo a implementação de diferentes camadas de visualização utilizando a mesma base de dados e fonte de informação. A arquitetura é adaptada de forma em que informações dos arquivos são armazenados em um banco de dados, os arquivos propriamente ditos em uma instância do CKAN e sistema de arquivos de um servidor. Foi implementado um sistema de APIs capaz de se comunicar com as duas primeiras camadas implementadas, realizando a conversão dos dados, persistindo os dados nas camadas dedicadas a armazenamento e realizar a comunicação para que todos os tipos de informações presentes nas duas camadas sejam utilizadas em conjunto, tornando a API responsável por automatizar serviços relacionados ao recebimento e compartilhamento de dados. As camadas abordadas diretamente no desenvolvimento do projeto são as camadas de Structuring e Collaborative

Figura 2 – Estrutura dos serviços da API.



Fonte: Corrêa, Corrêa & Silva (2014)

Os materiais utilizados para o desenvolvimento do sistema foram selecionados de forma a serem compatíveis com os padrões propostos para dados abertos, que envolvem softwares livres utilizados para o desenvolvimento,

permitindo o uso gratuito e ilimitado de tudo o que foi usado e com código aberto, permitindo que cada componente do sistema mantenha um código fonte acessível para todos os desenvolvedores, resultando assim em um software final com código aberto e disponível para uso livre.

O sistema implementado, se baseia na arquitetura proposta na Figura 2, e engloba toda a comunicação entre serviços de armazenamento, banco de dados, CKAN e APIs que permitem que o usuário realize o upload de um documento PDF inserindo informações sobre o mesmo, e de forma automática, o sistema deve converter o conteúdo do documento para CSV, armazenar informações no banco de dados, salvar o arquivo no CKAN e permitir acesso ao mesmo via CKAN ou API. A camada responsável por interagir diretamente com o usuário pode ser implementada em qualquer plataforma que permita interação com a API, incluindo smartphones, desktops, sistemas web entre outras plataformas. A camada dedicada ao usuário não é abordada na implementação deste sistema.

### 3.1 MATERIAIS

Foi utilizado a linguagem de programação orientada a objetos JAVA na versão disponibilizada pelo OpenJDK 8 e a IDE Eclipse para desenvolvimento do software. Para o desenvolvimento da REST API, um sistema web que se comunica por meio de requisições, respondendo apenas com documentos JSON, foi utilizado o framework Spring, um framework java que permite de forma simples converter dados para JSON e responder as requisições de usuários. Também foi utilizado um servidor Apache Tomcat e para conversão dos documentos PDF para CSV é utilizado a biblioteca TABULA PDF (TABULA, 2017). Uma instancia do banco de dados MariaDB é utilizado para armazenar dados dos arquivos e dos envios, e para persistência dos dados, é utilizado o framework Hibernate, pois facilita o armazenamento de grandes massas de dados no banco de dados. Um servidor Linux foi utilizado com a finalidade de testes e análise de resultado, porém o sistema é capaz de rodar em qualquer sistema operacional com capacidade de rodar a JVM, desde que o hardware atenda aos requisitos mínimos. Além disso, para armazenar os arquivos convertidos e compartilha-los, é utilizado uma instancia do CKAN, um portal com foco em compartilhamento de dados instalado e configurado em servidor separado.

### 3.2 MÉTODOS

Aqui são descritos os métodos que foram utilizados para a implementação dos métodos do sistema, baseando-se no método de camadas, os métodos de implementação partem do sistema de interface de API até o armazenamento e compartilhamento dos dados.

#### 3.2.1 api e json

Utilizando o sistema de RESTful services do Spring, é possível de forma automática converter classes JAVA para o formato JSON e enviar a informação como resposta a uma requisição HTML. Além de permitir o envio dos dados de uma classe, o Spring permite criar entidades do banco de dados, ou seja, uma classe preparada para receber e armazenar informações em um banco de dados utilizando o framework Hibernate, que é integrado ao sistema do Spring. O Spring ainda possibilita a transição direta das informações do banco de dados para a resposta no formato



JSON, sendo necessário uma implementação das classes de entidade e controladores compatível com o banco de dados.

Um exemplo de classe de controlador pode ser visto na Figura 3, e o objeto que ela transmite na Figura 4. Este controlador é responsável por transmitir ao usuário a versão do sistema, informação utilizada para manter sistemas compatíveis com a API atualizadas e funcionais de acordo com a versão da API.

Figura 3 – Código de implementação de um dos controladores da API.

```

1 package br.edu.ifsp.opendata;
2
3 import org.springframework.web.bind.annotation.RequestMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class VersionController {
8
9     private static final float version = 1.0F;
10
11     @RequestMapping("/version")
12     public Version version(String name) {
13         return new Version(version);
14     }
15 }

```

Fonte: Captura de tela realizada pelo autor (2017)

Figura 4 – Código de implementação da classe que será enviada como JSON.

```

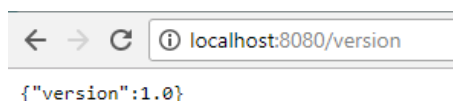
1 package br.edu.ifsp.opendata;
2
3 public class Version {
4
5     private final float version;
6
7     public Version(float version) {
8         this.version = version;
9     }
10
11     public float getVersion() {
12
13         return this.version;
14     }
15 }

```

Fonte: Captura de tela realizada pelo autor (2017)

Enquanto as classes JAVA de entidade funcionam como réplicas de tabelas do banco de dados, tornando-as ideais para receber, modificar e salvar os dados, as classes de controladores são responsáveis por transformar objetos JAVA para formato JSON e enviar as informações para consumo por meio de respostas da API. A requisição mais simples que pode ser realizada para a API é a chamada da versão, que pode ser acessada pela URL do servidor adicionando a palavra “version”. O resultado gerado pelo código disponível nas Figuras 3 e 4 pode ser observado na Figura 5

Figura 5 – Resposta da API sobre requisição "version".



```

{"version":1.0}

```

Fonte: Captura de tela realizada pelo autor (2017)

A chave “version” contém o valor 1.0, referente a versão da API. Como a estrutura do JSON permite múltiplas informações sendo transmitida por um envio de texto, qualquer linguagem de programação é capaz de trabalhar com os valores contidos no mesmo, necessitando apenas separar a chave do valor. Como uma REST API trabalha apenas com o envio de informações em formato de texto, grandes quantidades de dados podem ser transmitidas com um baixo consumo de internet e baixo processamento para envio, recebimento e uso dos dados, o que permite que, mesmo em lugares onde existam uma deficiência na internet, os dados possam ser acessados com pouco consumo de dados, ampliando ainda mais as possibilidades de uso em lugares com internet limitada.

O upload dos arquivos PDFs é realizado pela comunidade, e pode ser feita em qualquer plataforma desenvolvida para trabalhar com a API, pois acessando a URL de upload é possível via POST realizar o envio de arquivos para o servidor usando os formatos de transmissão de arquivos padrão do JAVA, o `InputStream` é uma classe java responsável pelo gerenciamento do fluxo de entrada de bytes no sistema, e é utilizado para realizar a leitura do arquivo que foi enviado via página HTML até o servidor, enquanto o `MultipartFile` é a forma como o servidor recebe envios html no formato MultiPart. O arquivo na linguagem JAVA se torna disponível para uso dentro do servidor graças ao envio via MultiPart e o `InputStream` capaz de acessar o conteúdo. As classes citadas, responsáveis por tratar dados de entrada e receber arquivos via web, possibilitam salvar o arquivo no disco no servidor. O arquivo recebido pela API, recebe com ele informações passadas pelo usuário, como nome do documento, local ao qual se refere, data de publicação e outras informações relacionadas ao conteúdo para serem utilizadas futuramente. Todas as informações são armazenadas no banco de dados MariaDB, em uma tabela desenvolvida em SQL para receber, armazenar os arquivos e armazena também o local e nome do arquivo, autor do envio, data de envio entre outras informações.

A infraestrutura de API desenvolvida fornece uma gama de recursos para buscar, baixar e adicionar arquivos para a base de dados. Considerando um servidor de testes local, é possível acessar aos serviços pela URL “localhost:8080/” concatenando o comando ao final. As possibilidades de uso podem ser vista na Tabela 3.2.1.

Tabela 2 – Métodos do sistema

Comando	Descrição
version	Comando responsável por responder com a versão da API.
upload/pdf	Utilizada para fazer upload de um documento PDF para ser convertido, salvo e enviado ao CKAN sendo utilizado com o método POST e recebendo como parâmetros um arquivo, a data a qual se referem os dados e o nome do documento, como parâmetros opcionais, temos a delimitação das áreas tabulares no pdf, as páginas que contém os dados tabulares e outras informações sobre como buscar e tratar os dados a serem convertidos. Os metadados de estado e cidade também podem ser enviados como parâmetro caso o documento contenha essas informações. O nome usuário responsável pelo upload também pode ser informado como parâmetro.

upload/csv	Utilizada para fazer upload de um documento CSV para que o mesmo seja salvo e enviado ao CKAN sendo utilizado com o método POST e recebendo como parâmetro um arquivo a data a qual se referem os dados e o nome do documento. Os metadados de estado e cidade também podem ser enviados como parâmetro caso o documento contenha essas informações. Neste caso o nome do usuário também pode ser informado como parâmetro.
buscar/nome	Utilizado pelo método GET, permite que uma busca seja realizada no banco de dados por um nome específico ou parte de um nome. Os parâmetros são o nome (name) a ser buscado, a quantidade de dados a ser exibida no JSON (limite), a pagina considerando a quantidade que será exibida (pagina) e a forma de ordenação dos dados (ordenado). Se nenhum nome for passado como parâmetro, o resultado será todos os dados ordenados pelo nome por ordem alfabética.
buscar/data	Utilizado pelo método GET, realiza uma busca no banco de dados por documentos com uma data específica. Os parâmetros são a data (data) a ser buscado, a quantidade de dados a ser exibida no JSON (limite), a pagina considerando a quantidade que será exibida (pagina) e a forma de ordenação dos dados (ordenado). Se nenhuma data for informada, todos os dados serão apresentados ordenados por data da mais antiga para a mais atual.
buscar/frequencia	Utilizado pelo método GET, permite que uma busca seja realizada no banco de dados por arquivos utilizados com mais frequência. Os parâmetros são a quantidade de dados a ser exibida no JSON (limite), a pagina considerando a quantidade que será exibida (pagina) e a forma de ordenação dos dados (ordenado).
buscar/mes	Utilizado pelo método GET, permite que uma busca seja realizada no banco de dados por um mês desconsiderando ano e dia. Os parâmetros são o mês (mes) em formato numérico, a quantidade de dados a ser exibida no JSON (limite), a pagina considerando a quantidade que será exibida (pagina) e a forma de ordenação dos dados (ordenado). Caso exista a ausência da informação do mês, todos os dados serão informados de forma ordenada por mês de janeiro até dezembro.

buscar/ano	Utilizado por meio do método GET, realiza uma busca no banco de dados por um ano desconsiderando mês e dia. Os parâmetros são o ano (ano) em formato numérico, a quantidade de dados a ser exibida no JSON (limite), a pagina considerando a quantidade que será exibida (pagina) e a forma de ordenação dos dados (ordenado). Caso exista a ausência do parâmetro ano, todos os dados serão informados de forma ordenada por ano em forma crescente.
buscar/dia	Utilizado com o método GET, realiza uma busca por um dia desconsiderando mês e ano. Os parâmetros são o dia (dia) em formato numérico, a quantidade de dados a ser exibida no JSON (limite), a pagina considerando a quantidade que será exibida (pagina) e a forma de ordenação dos dados (ordenado). Caso exista a ausência do parâmetro dia, todos os dados serão informados de forma ordenada por dia em forma crescente.
buscar/estado	Realiza uma busca por estados. Utiliza o método GET, e tem como parâmetros o estado (estado) que deve ser informado pela sigla, a quantidade de dados a ser exibida no JSON (limite), a pagina considerando a quantidade que será exibida (pagina) e a forma de ordenação dos dados (ordenado). Na ausência do parâmetro estado, todos os dados que pertencem a um estado serão exibidos ordenados por estados alfabeticamente.
buscar/cidade	Realiza uma busca por cidade. Utiliza o método GET, e tem como parâmetros a cidade (cidade) que deve ser informado pelo nome, a quantidade de dados a ser exibida no JSON (limite), a pagina considerando a quantidade que será exibida (pagina) e a forma de ordenação dos dados (ordenado). Na ausência do parâmetro cidade, todos os dados que pertencem a uma cidade serão exibidos ordenados por cidades alfabeticamente.
arquivo	Sendo um dos métodos mais importantes, é o responsável por informar tudo o que está disponível sobre determinado arquivo. Seu único parâmetro é o ID do documento, e é obrigatório. Caso o ID do documento não seja informado, todos os arquivos são exibidos ordenados pelo ID em ordem crescente.

Fonte:Produzido pelo autor (2017)

As chamadas relacionadas a busca de arquivos, e em caso de ausência dos parâmetros de limite e pagina, adotam os valores padrões adotados por REST API de limite=20 e pagina=0. Um caso exemplo onde a chamada “localhost:8080/buscar/mês/?limite=5&pagina=2” é feita, o resultado obtido é representado na Figura 6.

Figura 6 – Exemplo de resultado do uso da chamada “buscar/mes”.

```
{
  "back": "localhost/buscar/mês/?limite=5",
  "list": [
    {
      "id": "20",
      "nome": "Gastos de janeiro 2001"
    },
    {
      "id": "240",
      "nome": "Gastos de janeiro 2012"
    },
    {
      "id": "520",
      "nome": "Gastos de janeiro 2003"
    },
    {
      "id": "820",
      "nome": "Gastos de fevereiro 2006"
    },
    {
      "id": "30",
      "nome": "Gastos de fevereiro 2001"
    }
  ],
  "next": "localhost/buscar/mês/?limite=5&pagina=3"
}
```

Fonte: Captura de tela realizada pelo autor (2017)

A requisição realizada para a url “localhost:8080/arquivo/?ID=20” resultaria em algo como a Figura 7

Figura 7 – Exemplo de resultado do uso da chamada “arquivo”.

```
{
  "id":"20",
  "nome":"Gastos de fevereiro",
  "data":"18021997",
  "frequencia":"mensal",
  "cidade":"Campinas",
  "estado":["SP", "São Paulo"],
  "dataDeUpload":"25062000",
  "usuarioResponsavel":"Lucas",
  "nomeDoArquivoPDF":"GastosFev.pdf",
  "nomeDoArquivoCSV":"GAstosFev.csv",
  "url":"https://demo.ckan.org/api",
  "tamanho":"5mb"
}
```

Fonte: Captura de tela realizada pelo autor (2017)

### 3.2.2 Tabula

O processo de conversão utiliza a execução de uma versão do Tabula, um sistema desenvolvido em java que funciona por linha de comando e realiza a conversão arquivos PDF para formatos de documento CSV a partir do documento enviado pelo usuário. O sistema JAVA realiza comunicação com o Tabula por chamadas de linha de comando, fornecendo como parâmetro todas as informações necessárias para extrair os dados do documento PDF. Quando a conversão é completada, o CSV resultante é armazenado no sistema de armazenamento do servidor e no CKAN.

O Tabula é adicionada ao projeto JAVA, e desta forma, as chamadas do mesmo podem ser realizadas utilizando a própria linguagem. As chamadas em linha de comando realizadas em JAVA, permitem que o Tabula realize a tarefa de converter os dados, sem a necessidade do conhecimento do funcionamento do mesmo. Os comandos são executados utilizando um Runtime, classe do JAVA capaz de executar comandos diretamente no sistema operacional. Assim, com a documentação do tabula, a implementação da interface entre a linguagem JAVA e o Tabula é realizada a partir da execução de um comando, permitindo assim que o mesmo método seja utilizado para executar todas as ações, alterando apenas o comando a ser realizado. As chamadas de métodos variados podem ser realizadas pelo método exemplificado na Figura 9.

Figura 8 – Execução da chamada “-v” do Tabula que retorna a versão do mesmo.



```

1 import java.io.BufferedReader;
2 import java.io.InputStreamReader;
3
4 public class Teste {
5
6     public static void main(String[] args) {
7
8         try {
9
10            String comando = "java -jar /tabula.jar -v";
11            Process processo = Runtime.getRuntime().exec(comando);
12
13            BufferedReader saida = new BufferedReader(new
14                InputStreamReader(processo.getInputStream()));
15
16            String s = null;
17            while ((s = saida.readLine()) != null) {
18                System.out.println(s);
19            }
20
21        }
22        catch (Exception e) {
23            e.printStackTrace();
24        }
25    }
26 }

```

tabula 1.0.1 (c) 2012-2017 Manuel Arístarán

Fonte: Captura de tela realizada pelo autor (2017)

Com isso, todos os métodos e parâmetros podem ser passados ao Tabula por meio de uma String, permitindo que dinamicamente, os parâmetros necessários sejam indicados ao Tabula. Todos os comandos do tabula estão disponíveis em sua página do GitHub.

### 3.2.3 ckan

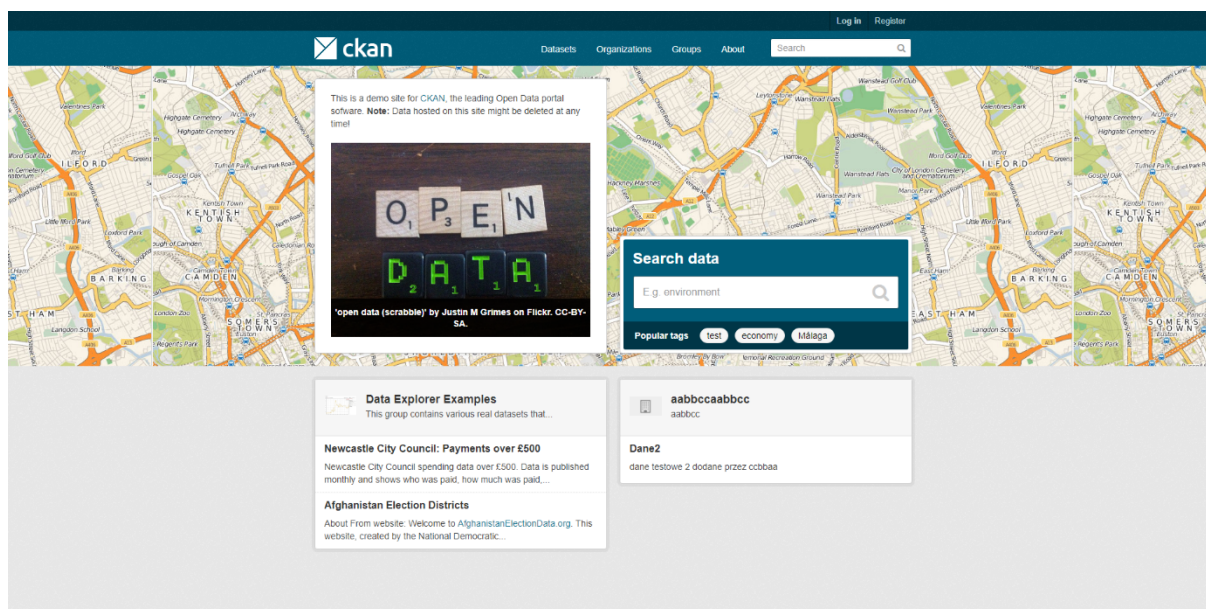
O documento já convertido, é enviado ao CKAN, uma plataforma focada no compartilhamento de dados abertos. Essa plataforma permite armazenar, arquivos e metadados dos arquivos, fornecendo assim uma forma simples de compartilhar, baixar e usar os dados. O MariaDB, permite que os dados sejam transitados para sistemas e portais que utilizem a API, enquanto o CKAN já é dedicado ao usuário final, que pretende usar os dados e acessar suas informações diretamente na página do CKAN. As informações antes armazenadas no banco de dados, são enviadas ao CKAN juntamente com o arquivo, via CKAN API, realizando assim o armazenamento público dos dados e permitindo compartilhamento de livre acesso para todos. O arquivo CSV é enviado ao servidor do CKAN para uma pasta específica, onde são armazenados todos os arquivos disponíveis, e adicionado via Storage API do próprio CKAN pelo método POST, onde são passados por parâmetro o tamanho do documento, o tipo, o nome para acesso no servidor do CKAN, a data da última modificação do arquivo e a URL do documento no servidor do sistema. Esses dados são todos armazenado no banco de dados, incluindo a URL de acesso ao arquivo no CKAN, e servidor local, resultando em duas URLs que serão utilizadas para baixar o arquivo via REST API.

Além de permitir armazenamento e gerenciamento dos dados, o CKAN ainda conta com uma interface gráfica baseada em plataformas web, e foi desenvolvida para uso direto do usuário final. O acesso é irrestrito e com facilidades fornecidas pelo próprio CKAN, podendo também se utilizar da API do CKAN para utilizar apenas os dados em qualquer outro sistema, independente de linguagem, sistema operacional, ou plataforma. O CKAN ainda permite buscas avançadas com dados variados, buscas para permitir maior precisão nos resultados, download de grandes massas de dados em poucas etapas, além de outras ferramentas de edição e organização dos dados.

A tela inicial do CKAN, onde o usuário é apresentado diretamente ao mecanismo de busca da plataforma e recebe

algumas informações complementares, e além de ser simples e funcional, ainda oferece implementações de buscas rápidas e eficientes, para garantir que o usuário chegue facilmente até o dado que procura. A interface é baseada em um caixa de busca que possibilita que o usuário navegue por toda a base de dados, assim como na Figura ??.

Figura 9 – Pagina inicial do CKAN



Fonte: Captura de tela do site demo.ckan.com tirado pelo autor (2017).



## 4 TESTES

Os testes foram realizados utilizando servidor local, e um servidor Microsoft Azure rodando o sistema operacional Linux Ubuntu, disponibilizado para testes. A instancia do CKAN utilizada, foi a fornecida pelo site [demo.ckan.org](http://demo.ckan.org), plataforma gratuita e livre para uso, que fornece uma instancia funcional do CKAN com armazenamento temporário.

Os testes realizados inicialmente foram testes modulares, ou seja, testes de partes separadas do sistema, onde foram realizados testes de envios de arquivos por meio da API, conversões de documentos PDFs variados, requisições de arquivos e downloads dos mesmos por meio da API do sistema, e downloads do CKAN. Todos os testes modulares foram realizados aproximadamente 10 vezes, para garantir fidelidade dos resultados e impedir que testes viciados gerem resultados incorretos. Os testes de uso do sistema foram realizados com envios e conversões de um único PDF, e também com massas de 30 PDFs simultâneos, onde a quantidade de conversões simultânea chegou a 70 arquivos. Os resultados de ambos os testes foram documentos CSV com o conteúdo fiel ao do PDF, levando em consideração que os PDFs utilizados para os testes eram documentos considerados como ideais para conversão.

Testes realizados com arquivos incompatíveis e PDFs contendo caracteres incompatíveis, resultam em um arquivo nulo ou defeituoso, que é removido do sistema sem realizar alterações no banco de dados e no CKAN.

Uma massa de testes de integração e de caso de uso foram realizados, ou seja, o processo partindo do upload do documento PDF até a busca pelo arquivo e download do mesmo por meio da API do sistema e exibição. Testes onde grandes massas de arquivos foram enviadas e baixadas utilizando a API do CKAN e a do sistema em si, além disso, os arquivos foram adquiridos pela interface web do próprio CKAN. Em ambos os casos não foram encontrados erros em nenhuma etapa, de forma em que 50 arquivos foram baixados de um único pacote do CKAN, e os mesmos 50 arquivos foram recuperados facilmente pela API desenvolvida, o armazenamento era realizado com sucesso e todos os dados mantidos intactos e fieis aos documentos originais.

## 5 RESULTADOS

A massa de testes utilizada para testar todo o processo do serviço oferecido pelo software, foi de 90 PDFs de dados tabulares governamentais do estado de São Paulo, onde foram utilizados 30 considerados ideais, 30 inapropriados e 30 incompatíveis com o método de conversão de dados. O tempo de upload e de upload de arquivos variou entre 1 e 5 segundos, tendo como principais variáveis que influenciam no tempo o tamanho dos dados e qualidade da conexão, enquanto o tempo necessário para o processo de conversão depende apenas do tamanho do arquivo e quantidade de dados.

Na massa de arquivos utilizada para testes, os documentos considerados como ideais obtiveram sucesso em 100% das conversões realizadas. Os dados inapropriados, resultaram em problemas causados por caracteres incompatíveis ASCII e UTF, causando erros nas conversões dos dados e tendo como única forma de correção a reimplantação da biblioteca ou utilização de uma possível alternativa. Em casos de colunas ou linhas onde existiam células mescladas, os dados resultantes não eram fieis aos apresentados. Como exemplo, a Figura 10, uma tabela com células mescladas submetida ao processo de conversão do Tabula, resultara na perda de dados, onde as tabelas mescladas não são consideradas, resultando em algo como a Figura 11. Os dados resultantes são incompletos, e desta forma, é obtido um documento inutilizável e que não possui os mesmos dados do documento PDF original.

Tabela com células mescladas

Figura 10 – Tabela com células mescladas.

1	2	
1		
1	3	
1	4	
1		

Fonte: Captura de tela realizada pelo autor (2017)

Figura 11 – Tabela com células mescladas após conversão pelo Tabula.

1	2	
1		
1	3	
1	4	
1		

Fonte: Captura de tela realizada pelo autor (2017)

Os dados incompatíveis, não utilizam o padrão de tabelas dos formatos PDFs, utilizando como forma de exibição a imagem de uma tabela salva em um documento. O resultado disso é incompatibilidade total e impossibilidade de extração dos dados.

Não foram encontrados nenhum tipo de falha nos uploads e nem na transferência de dados entre os servidores do sistema e os servidores do CKAN, que são realizados por meio da API do CKAN mantendo assim informações integras aos dados convertidos. O CKAN permite acesso irrestrito e sem necessidade de cadastros ou informações adicionais, permitindo assim, compartilhamento amplo e totalmente compatível com dados abertos.

Complementando o CKAN, a API do sistema permite acesso irrestrito aos documentos contidos no CKAN além da própria interface CKAN que fornece os arquivos diretamente aos usuários finais, atingindo assim o objetivo de compartilhar os dados de forma simples e padronizada.

## 6 CONCLUSÃO

Na era da tecnologia e da informação, vivemos em um importante cenário onde o acesso a informação é essencial. Por lei temos direito ao acesso a documentos governamentais de forma simples e direta, o que é positivo em vários aspectos, porém, ainda é possível notar falhas em etapas do processo. As deficiências do processo, podem ser catastróficas em relação ao uso dos dados, porém, com o uso do software desenvolvido, é possível atingir um resultado mais próximo do proposto pelos padrões de dados abertos, garantindo assim em um resultado próximo do ideal garantindo formato compatível, qualificado e compartilhamento acessível e padronizado.

As ferramentas com mesmo objetivo de converter PDFs para formatos de fácil uso, normalmente oferecem preços elevados, e o próprio tabula oferece um uso muito dificultado, de forma em que um usuário comum dificilmente conseguiria utilizar o sistema, sem conhecimento prévio de programação. Neste sentido, a extração específica de dados tabulares contidos em PDF não é uma tarefa trivial de ser alcançada por qualquer usuário. Com a implementação dessa API, qualquer usuário poderá valer-se dos serviços oferecidos de modo simples e rápido.

Como resultado do uso do software, é possível tornar os documentos compatíveis com leitura por máquina e assim permitindo não apenas o trabalho com grandes quantidades de dados sem grande esforço humano, mas também armazenamento seguro e organizado dos dados e possibilitando sistemas automatizados de utilizarem informações e atingindo assim todas as metas necessárias para um documento totalmente compatível com dados abertos. Além disso, o compartilhamento padronizado e o sistema de APIs, proporcionam amplo acesso para desenvolvedores com interesse em consumir os dados e usuários finais com o mesmo objetivo.

O uso do sistema não se restringe apenas para usuários comuns, é esperado também que o próprio governo implemente uma instância e utilize o sistema, com objetivo de utilizar o mesmo como meio de transição, até que o processo de geração de dados abertos atinja o ideal proposto pela transparência governamental. A facilidade do uso padroniza formato de arquivo e compartilhamento compatível, além de outras ferramentas e facilidades.

O sistema aberto permite não apenas uso governamental, mas também qualquer outro uso, e o código aberto permite implementações para fins variados de acordo com os interesses do desenvolvedor. O acesso a todo o código é irrestrito, e pode ser realizado por meio da plataforma GIT HUB, permitindo implementações distintas e livre para modificações e uso sem nenhum tipo de impedimento ou dificuldade facilitando futuras implementações e melhorias.

## REFERÊNCIAS

BENNETT, D.; HARVEY, A. **Publishing Open Government Data**. W3C, 2009. Disponível em: <<https://www.w3.org/TR/gov-data/>>.

BRASIL, OPEN KNOWLEDGE. **Índice de dados abertos para o Brasil**. 2017. Disponível em: <<http://dapp.fgv.br/wp-content/uploads/2017/04/IndiceDadosAbertosBrasil2017.pdf>>.

CORRÊA, Andreiuid Sheffer; CORRÊA, Pedro Luiz Pizzigatti; SILVA, Flávio Soares Corrêa da. Transparency portals versus open government data: an assessment of openness in brazilian municipalities. In: ACM. **Proceedings of the 15th Annual International Conference on Digital Government Research**. [S.l.], 2014. p. 178–185.

FOUNDATION, OPEN KNOWLEDGE. **OPEN DATA HANDBOOK**. Open KNOWLEDGE INTERNATIONAL, 2014. Disponível em: <<http://opendatahandbook.org>>.

INTERNATIONAL, OPEN KNOWLEDGE. **GlobalOpen Data Index**. 2016. Disponível em: <<https://index.okfn.org/place/>>.

MANOCHA, Ian. **On the Road to Open Data**. 2011. Disponível em: <<http://www.idgconnect.com/blog-abstract/263/ian-manocha-uk-on-road-open-data>>.

RIBEIRO, Claudio Jose Silva. Dados abertos governamentais (open government data): instrumento para exercício de cidadania pela sociedade.

TABULA. **tabula-java**. nov 2017. Disponível em: <<https://github.com/tabulapdf/tabula-java>>.

TRANSPARÊNCIA, Ministério da. **Painel de monitoramento de dados abertos**. nov 2017. Disponível em: <<http://paineis.cgu.gov.br/dadosabertos/index.htm>>.