



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
BACHARELADO EM ENGENHARIA DE SOFTWARE

MAURICIO JOSÉ DE LIMA

UM SERVIÇO PARA O ACESSO A DADOS ABERTOS EM NUVEM

**QUIXADÁ
2014**

MAURICIO JOSÉ DE LIMA

UM SERVIÇO PARA O ACESSO A DADOS ABERTOS EM NUVEM

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Engenharia de Software da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: computação

Orientador Prof. Dr. Flávio Rubens de Carvalho Sousa

**QUIXADÁ
2014**

L699s Lima, Mauricio José de
Um serviço para o acesso a dados abertos em nuvem / Mauricio José de Lima. – 2014.
37 f. : il. color., enc. ; 30 cm.

Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Bacharelado em Engenharia de Software, Quixadá, 2014.
Orientação: Prof. Dr. Flávio Rubens de Carvalho Sousa
Área de concentração: Computação

1. Computação em nuvem 2. Software livre 3. Sistemas de computação - Tecnologia da Informação
I. Título.

MAURICIO JOSÉ DE LIMA

UM SERVIÇO PARA O ACESSO A DADOS ABERTOS EM NUVEM

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Engenharia de Software da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: computação

Aprovado em: _____ / junho / 2014.

BANCA EXAMINADORA

Prof. Dr. Flávio Rubens de Carvalho Sousa (Orientador)
Universidade Federal do Ceará-UFC

Prof. MSc. Regis Pires Magalhaes
Universidade Federal do Ceará-UFC

Prof. Dr. Lincoln Souza Rocha
Universidade Federal do Ceará-UFC

AGRADECIMENTOS

Aos meus pais, que sempre me deram total apoio e sempre fizeram de tudo para que chegasse até aqui.

Aos meus familiares pelo apoio.

Aos meus amigos de graduação, em especial a dois amigos Klairton e Bruno que sempre procuraram me ajudar e serviram de inspiração em grande parte dessa caminhada.

A minha namorada Dayse que sempre me incentivou e nunca me deixou desanimar durante o desenvolvimento desse trabalho.

Aos meus amigos de tantos anos Romário e Leandro pelo apoio.

E ao professor Flávio, que como Professor, Coordenador e Orientador sempre me motivou e guiou não só para a conclusão desse trabalho, mas também para a conclusão desse curso.

Muito Obrigado.

"Sonhos determinam o que você quer. Ação determina o que você conquista."
(Aldo Novak)

RESUMO

Com a criação da Lei nº 12.527/2011 ou Lei de Acesso à Informação, o uso da política de dados abertos surge como uma forma de melhorar comunicação entre o governo e a população em um país com tanta diversidade cultural como o Brasil, contudo a falta de um serviço que auxilie na disponibilização desses dados torna essa tarefa um pouco complicada. Esse trabalho propõe um serviço para auxiliar no armazenamento e disponibilização de dados abertos juntamente com a tecnologia da computação em nuvem, buscando explorar as vantagens oferecidas por essa tecnologia. Dessa forma, os dados disponibilizados pelo serviço estariam prontos para serem disponibilizados e acessados de qualquer lugar e por qualquer dispositivo que esteja conectado a internet. Outro benefício que o uso da tecnologia em nuvem proporcionou para esse trabalho foi o uso da escalabilidade vertical, ou seja, a capacidade das máquinas individuais pode ser ajustada caso o serviço esteja lento ou degradado. A avaliação desse trabalho foi feita por meio de um estudo de caso onde foi escolhida uma base de dados para ser disponibilizada pelo serviço e criada uma aplicação cliente que utilizasse esse serviço para consumir os dados disponibilizados.

Palavras chave: Computação em Nuvem. Dados Abertos.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 – Arquitetura geral do serviço..... | 20 |
| Figura 2 – Diagrama de Sequencia do Serviço..... | 21 |
| Figura 3 – Tela de Consultas | 24 |
| Figura 4 – Serialização das mensagens de resposta | 24 |
| Figura 5 – Requisição ao banco de dados com a consulta desejada..... | 25 |
| Figura 6 – Tela inicial..... | 26 |
| Figura 7 – Tela da Documentação | 26 |
| Figura 8 – Documentação expandida do método distrito | 27 |
| Figura 9 – Tela Consultas | 27 |
| Figura 10 – Tela consulta da entidade Escola | 28 |
| Figura 11 – Aplicação Cliente | 29 |

SUMÁRIO

| | |
|---|----|
| 1 INTRODUÇÃO..... | 10 |
| 1.1 Objetivo geral | 11 |
| 1.2 Objetivos específicos | 12 |
| 2 REVISÃO BIBLIOGRÁFICA | 12 |
| 2.1 Computação em Nuvem..... | 12 |
| 2.2 Dados Abertos..... | 14 |
| 3 PROPOSTA..... | 17 |
| 3.1 METODOLOGIA | 18 |
| 3.1.1 Definir os requisitos do sistema | 18 |
| 3.1.2 Analisar e definir as tecnologias que serão utilizadas no desenvolvimento do serviço | 18 |
| 3.1.3 Analisar e definir a base de dados que será utilizada no estudo de caso..... | 18 |
| 3.1.4 Desenvolvimento do serviço | 18 |
| 3.1.5 Realizar o estudo de caso..... | 19 |
| 3.2 A SOLUÇÃO..... | 19 |
| 3.3 ARQUITETURA | 20 |
| 3.4 IMPLEMENTAÇÃO E TECNOLOGIAS | 21 |
| 3.4.1 DEFINIÇÃO DAS TECNOLOGIAS..... | 22 |
| 3.4.2 IMPLEMENTAÇÃO DO SERVIÇO..... | 23 |
| 4 ESTUDO DE CASO | 25 |
| 4.1 CENÁRIO DE USO | 25 |
| 4.2 FLUXO DE EXECUÇÃO | 25 |
| 4.3 APLICAÇÃO CLIENTE..... | 28 |
| 5 CONSIDERAÇÕES FINAIS | 29 |
| REFERÊNCIAS | 31 |

1 INTRODUÇÃO

Diante da dificuldade de comunicação entre o governo e a população em um país com tanta diversidade cultural, social e educacional, surge o conceito de dados abertos como um grande avanço para uma maior participação da população no desenvolvimento da sociedade (TRINDADE et al., 2012). Contudo, boa parte das informações do serviço público é disponibilizada com uma organização estrutural de difícil entendimento para o usuário comum (DINIZ, 2010). Devido ao grande desenvolvimento da tecnologia na sociedade moderna, surgiu também a necessidade de ser ter uma nova infraestrutura para os serviços da Internet que dê suporte ao grande aumento do número de usuários conectados, multiplicando assim os dados armazenados, possibilitando o uso desses recursos computacionais em qualquer lugar desde que se tenha algum dispositivo com acesso a Internet (RODRIGUES; SANTANA, 2010).

Com a computação em nuvem, que permite disponibilizar serviços de tecnologia da informação (TI) sob demanda e com pagamento baseado no uso (pay-per-use), um número maior de sistemas têm sido desenvolvidos para disponibilizar informações a sociedade. A elasticidade da nuvem permite que os utilizadores dessa infraestrutura usem os recursos que são oferecidos na quantidade que forem necessários, aumentando ou diminuindo o seu desempenho computacional dinamicamente (TAURION, 2009, p.2-3).

A computação em nuvem implica no uso de grandes repositórios de recursos facilmente acessíveis através da Internet (ENDLER et al., 2011). Ela oferece serviços para diferentes públicos-alvo, desde o usuário comum que utiliza o serviço de uma forma simples como o armazenamento de pequenos documentos até empresas de modo geral que utilizam essa tecnologia em toda a sua infraestrutura. Por meio da computação em nuvem, todos os recursos podem ser consumidos diretamente na nuvem, sendo entregues aos seus diferentes utilizadores sob demanda (SOUSA et al., 2009).

A tecnologia de computação em nuvem vem sendo amplamente utilizada. A cada dia, mais instituições e empresas migram suas aplicações para o ambiente em nuvem devido a facilidade que se tem em acessar esses dados de forma simples e de qualquer lugar. Com isso o termo computação em nuvem deixou de ser uma tendência dentro do universo de Tecnologia da Informação (TI) e passou a ser essencial (LOPES; JUNIOR, 2012).

Com a utilização da computação em nuvem, o acesso e a disponibilidade de dados abertos para a população vai se tornar mais fácil, devido aos grandes benefícios que essa tecnologia oferece como custo reduzido, disponibilidade e flexibilidade. Esses benefícios fazem desse modelo uma ótima solução para quem quer disponibilizar seus dados para o acesso da população.

Existem alguns trabalhos que abordam tanto a disponibilização de dados abertos em nuvem. O trabalho de Trindade et al. (2012) tem como objetivo a construção de uma aplicação Web que facilite o entendimento das informações da Câmara Municipal de São Paulo, que já são disponibilizadas através de dados abertos no formato XML (Extensible Markup Language), mas que por causa do formato em que essas informações se encontram a legibilidade para a análise dessas informações se torna mais difícil para pessoas que não tem conhecimento sobre esse formato.

O trabalho de Aranda-Andújar et al. (2012) apresenta uma plataforma chamada Amada, sua função é o armazenamento de dados na WEB com base na infraestrutura de nuvem da Amazon Web Services(AWS), esses dados são documentos em formato XML e grafos de formato RDF. A plataforma utiliza a abordagem Software como Serviço (SaaS), permitindo que os usuários dessa plataforma possam armazenar e consultar grandes volumes de dados da Web. Já o trabalho de Breitman et al. (2012) discute sobre o estado atual da publicação de Dados Governamentais Abertos (DGA) no Brasil e sobre alguns desafios que devem ser cumpridos para que essa abordagem de publicação dos DGA funcione de fato.

Este trabalho propõe um serviço para o acesso a dados abertos em nuvem, onde qualquer empresa poderá disponibilizar seus dados utilizando esse serviço desde que a mesma possua uma base de dados pronta para ser usada e disponibilizada por esse serviço. Esse serviço vai disponibilizar as informações abertas de uma forma que qualquer usuário, possa entender e utilizar os dados que estão disponibilizados pelo serviço da forma que desejarem. Esse serviço irá explorar algumas vantagens que a computação em nuvem oferece, como na facilidade e na disponibilidade que ela oferece para quem usa esse tipo de serviço.

1.1 Objetivo geral

- Desenvolver um serviço para prover acesso a dados abertos que utilize o ambiente de computação em nuvem.

1.2 Objetivos específicos

- Projetar um serviço de dados abertos utilizando o ambiente de computação em nuvem.
- Desenvolver uma aplicação cliente para acessar o serviço.
- Selecionar uma base de dados aberta para ser utilizada no estudo de caso.
- Realizar um estudo de caso utilizando a aplicação cliente e verificar se o serviço possibilita o acesso aos dados armazenados no próprio serviço.

2 REVISÃO BIBLIOGRÁFICA

Na revisão bibliográfica do projeto serão abordados os conceitos utilizados em seu desenvolvimento. Na primeira seção será definido o que é a Computação em Nuvem e na segunda seção será definido o conceito de dados abertos.

2.1 Computação em Nuvem

Computação em nuvem propõe a integração de diversos modelos tecnológicos para o provimento de infraestrutura de hardware, plataformas de desenvolvimento e aplicações na forma de serviços sob demanda com pagamento baseado em uso (BUYA, 2009). Neste novo paradigma de utilização de recursos computacionais, clientes dispensam a administração de uma infraestrutura própria e utilizam serviços oferecidos por terceiros, delegando responsabilidades e assumindo custos estritamente proporcionais à quantidade de recursos utilizados.

Os serviços em nuvem apresentam diversas vantagens para os usuários, tais como: previsibilidade e custos mais baixos, proporcionais à qualidade de serviço (QoS) e cargas de trabalho reais; complexidade técnica reduzida, graças a interfaces de acesso unificado e administração simplificada; e elasticidade e escalabilidade, proporcionando a percepção de recursos teoricamente infinitos. Por outro lado, o provedor tem que garantir a percepção de recursos infinitos sob cargas de trabalho dinâmicas e minimizar os custos operacionais associados a cada usuário (SOUSA, et al 2010). Contudo, a nuvem apresenta uma variabilidade de desempenho bastante elevada, principalmente devido à heterogeneidade de hardware, virtualização e compartilhamento de recursos, dificultando o desenvolvimento de soluções com QoS na nuvem (SCHAD, 2010).

Apesar das limitações de rede e segurança, as soluções em nuvem devem fornecer um elevado desempenho, além de serem flexíveis para se adaptarem diante de uma

determinada quantidade de requisições. Como, em geral, os ambientes de computação em nuvem possuem acesso público, torna-se imprevisível e variável a quantidade de requisições realizadas, dificultando fazer estimativas gerais e fornecer garantias de QoS (SCHAD,2010). Essas garantias de qualidade são definidas entre o provedor do serviço e cada usuário e expressas por meio de um acordo de nível de serviço (SLA), que consiste de contratos que especificam um nível de desempenho que deve ser atendido e penalidades em caso de falha (SOUSA, et al 2012).

A elasticidade é ponto chave para implementar serviços com QoS para nuvem, pois permite adicionar ou remover recursos, sem interrupções e em tempo de execução para lidar com a variação da carga (COSTA, et al 2011; GALANTE; DE BONA, 2012; COUTINHO, et al 2013). De acordo com Mell e Grance (2011), estes recursos podem ser adquiridos de forma rápida, em alguns casos automaticamente, caso haja a necessidade de serem escalados com o aumento da demanda, e liberados, na retração dessa demanda. Para os usuários, os recursos disponíveis para uso parecem ser ilimitados e podem ser adquiridos em qualquer quantidade e a qualquer momento.

Para implementar a elasticidade, as principais formas utilizadas são técnicas de replicação e redimensionamento de recursos (COUTINHO, et al 2013). Técnicas de replicação são utilizadas para implementar a elasticidade do ajuste automaticamente da quantidade de número de réplicas para a carga de trabalho atual (SOUSA; MACHADO, 2012). O redimensionamento permite a ajuste da quantidade de recursos de acordo com a carga de trabalho (REGO, P. A. L., et al 2011).

Por outro lado, os ambientes de nuvem são inerentemente grandes, complexos, heterogêneos, altamente dinâmicos e os provedores devem tratar questões de QoS, dificultando o desenvolvimento de soluções elásticas na nuvem (ZHANG; CHENG; BOUTABA, 2010). A nuvem possui diferenças em relação aos sistemas tradicionais, tais como intervenção humana limitada, alta alternância na carga de trabalho e uma variedade infraestruturas compartilhadas. Na maioria dos casos, não haverá administradores de sistemas em nuvem ou de sistemas para ajudar os desenvolvedores, fazendo com que a solução seja automatizada ao máximo (SOUSA, et al 2010).

2.2 Dados Abertos

O conceito de dados abertos passou a ser mais conhecido e utilizado com a criação da Lei nº 12.527/2011, também conhecida como Lei de Acesso à Informação, essa lei permite qualquer cidadão, sem necessidade de apresentar uma justificativa, solicite dados e informações a qualquer órgão ou entidade pública dos poderes Executivo, Legislativo e Judiciário, além do Ministério Público, nas esferas Federal, Estadual e Municipal. Segundo definição da Cartilha Técnica para Publicação de Dados Abertos no Brasil, Dados Abertos ou Dados Governamentais Abertos (DGA) são dados ou informações públicas que são publicadas na Web, seguindo critérios que fazem com que esses dados possam ser reutilizados e usados para o desenvolvimento de aplicativos para toda a sociedade.

Para o W3C (Consórcio World Wild Web), dados governamentais abertos é a disponibilização de informações governamentais representadas em formato aberto e acessível de modo que possam ser reutilizadas, misturadas com informações de outras fontes, criando novos significados.

A definição de dados abertos segue três leis criadas por Eaves (2009) que são consideradas pela W3C:

- Se o dado não pode ser localizado na Web, esse dado não existe;
- Se o dado não estiver aberto e disponível em um formato acessível por qualquer máquina, esse dado não pode ser utilizado;
- Se a reutilização desses dados não for permitida por algum motivo legal, esse dado não é útil.

Vaz et al. (2010), define Dados Governamentais Abertos como públicos e pertencentes ao cidadão, que deveria ter acesso sem restrições as informações governamentais. Para ele, o grande crescimento no uso das tecnologias da informação, onde a internet, ferramentas e os formatos abertos facilitam a disponibilização e o acesso a essas informações.

De acordo com um grupo de especialistas denominado OpenGovData a implementação de DGA devem ser guiadas através de oito princípios criados por esse grupo em 2007. No quadro 1 estão listados os princípios e suas definições sobre como os DGA devem ser:

Quadro 1 – Os oito princípios dos dados abertos criados pelo grupo OpenGovData.

| Princípios | Definição |
|------------------------------------|--|
| Completos | Todos os dados públicos estão disponíveis e não limitados por restrições de segurança ou acesso. |
| Primários | Os dados são apresentados iguais aos que foram coletados, sem nenhuma modificação. |
| Atuais | Os dados são disponibilizados tão rapidamente quanto a necessário à preservação do seu valor. |
| Acessíveis | Os dados são disponibilizados para o maior alcance de usuários, possibilitando sua utilização para um maior número de finalidades. |
| Compreensíveis por máquinas | Os dados são estruturados de modo que possibilite o processamento automatizado. |
| Não discriminatórios | Os dados são disponibilizados para todos, sem exigências de cadastro. |
| Não proprietários | Os dados são disponibilizados em formato que nenhuma entidade tenha o controle exclusivo sobre o mesmo. |
| Livres de licenças | Os dados não podem ter restrições de direito autoral, patente, entre outros. |

Fonte: Elaborado pelo autor.

Segundo (DADOS, 2013), apesar desses princípios terem sido pensados para os DGA, eles também podem ser aplicados a dados abertos de modo geral.

Para Diniz (2010), a disponibilização de dados governamentais abertos admite que as informações sejam usadas do modo desejado pelo interessado, de forma que essas informações possam ser manipuladas a fim de agregar mais valor aos dados, com isso, os usuários das informações do serviço público vão conseguir encontrar, acessar, entender e utilizar os dados públicos de acordo com o seu interesse com mais facilidade.

Dados (2013) por meio da Cartilha Técnica para Publicação de Dados Abertos no Brasil define que para um dado ser considerado aberto ele “deve estar disponível em um formato de especificação aberta, não proprietário, e estruturado que possibilite seu uso sem restrições através da Web”. Dados (2013) ainda alerta sobre um erro comum citado na

cometido por diversas instituições que é a publicação desses dados no formato PDF, que devido ao seu formato não estruturado a reutilização desses dados fica mais difícil.

No quadro 2 são listados alguns formatos não proprietários para dados abertos e suas características:

Quadro 2 – Formatos para dados abertos definidos na Cartilha Técnica para Publicação de Dados Abertos no Brasil.

| Formatos | Características |
|--|--|
| JSON (JavaScript Object Notation) | <ul style="list-style-type: none"> • Possui um padrão aberto de estruturação de dados baseado em texto e legível por humano. • A serialização em JSON é muito simples e possibilita a serialização de estrutura de objetos complexos, como listas e subpropriedades. • Está se tornando o padrão mais utilizado para a integração de dados entre repositórios e frameworks. |
| XML (Extensible Markup Language) | <ul style="list-style-type: none"> • Conjunto de regras para codificar documentos com estrutura hierárquica e possui um formato legível por máquina. • Baseado em texto e tem como objetivo a simplicidade, extensibilidade e usabilidade. • Muito utilizado como formato de troca de dados nos Web Services SOAP. |
| CSV (Comma-Separated Values) | <ul style="list-style-type: none"> • Simples codificação: cada linha do arquivo representa uma linha na tabela e as colunas são separadas por vírgula. • Recomendado para a representação de estrutura de dados mais simples, onde não existem subpropriedades ou listas. • Gera um arquivo menor e mais leve para o processamento. |

| | |
|---|---|
| ODS (Open Document Spreadsheet) | <ul style="list-style-type: none"> • Possui um formato não proprietário de arquivo baseado em XML. • É muito flexível, possibilitando a manipulação de diversos tipos de dados. |
| RDF (Resource Description Framework) | <ul style="list-style-type: none"> • Modelo de dados estruturado em grafos e possui diversos formatos de serialização. • Facilita a integração de dados e a interligação de dados entre diferentes fontes de dados. |

Fonte: Elaborado pelo autor.

A publicação de dados abertos não envolve apenas a publicação dos dados em si, mas também a publicação dos seus metadados. Dados (2013) por meio da Cartilha Técnica para Publicação de Dados Abertos no Brasil define metadados como “informações que possibilitam organizar, classificar, relacionar e inferir novos dados sobre um conjunto de dados”. No quadro 3 serão listados alguns metadados obrigatórios para a publicação de dados abertos e suas respectivas definições.

Quadro 3 – Metadados obrigatórios segundo a Cartilha Técnica para Publicação de Dados Abertos no Brasil.

| Metadados | Definição |
|--------------------------|--|
| Título | Nome do conjunto de dados. |
| Descrição | Pequena explicação sobre os dados. |
| Catálogo Origem | URL do órgão onde está publicado os dados. |
| Órgão Responsável | Nome e sigla do órgão responsável pela publicação dos dados. |

Fonte: Elaborado pelo autor.

A Cartilha Técnica para a Publicação de Dados Abertos no Brasil ainda sugere alguns metadados desejáveis. Porém pelo fato de serem desejáveis seus conceitos não serão abordados nesse trabalho.

3 PROPOSTA

Com a criação da Lei nº 12.527/2011 ou Lei de Acesso à Informação, lei que da o direito ao cidadão brasileiro de ter acesso a dados de qualquer serviço público do país ou fundações sem fins lucrativos que utilizam recursos públicos desde que esses dados não

estejam classificados como sigilosos. Essa lei se aplica a toda esfera pública, desde o serviço público municipal até o serviço público federal, com isso, surgiu a necessidade de termos serviços que auxiliem na disponibilização desses dados e que facilite a utilização desses dados pelos usuários.

Esse trabalho propõe um serviço Web para auxiliar na disponibilização e acesso a dados abertos por meio de uma API que permitirá ao seu utilizador usar esses dados da forma que desejar.

3.1 METODOLOGIA

Nessa sessão serão explicados os passos realizados durante o desenvolvimento desse projeto.

3.1.1 Definir os requisitos do sistema

Nessa fase, foram definidas as funcionalidades do serviço, os seus requisitos, e como será o seu funcionamento, definindo os parâmetros de entrada do serviço e as informações de saída, levando em consideração algumas restrições do serviço.

3.1.2 Analisar e definir as tecnologias que serão utilizadas no desenvolvimento do serviço

Nessa fase do projeto foram definidas as tecnologias que vão ser utilizadas durante o desenvolvimento do serviço. A escolha foi definida por meio de algumas comparações entre as tecnologias e ferramentas atuais no campo de desenvolvimento de serviços Web e também levando em consideração o conhecimento do autor sobre essas tecnologias.

3.1.3 Analisar e definir a base de dados que será utilizada no estudo de caso

Nessa fase do projeto foi definida a base de dados utilizada durante o estudo de caso do serviço desenvolvido por esse trabalho. A escolha da base de dados foi guiada por comparações entre base de dados abertas existentes e também levando em consideração a preferência do autor.

3.1.4 Desenvolvimento do serviço

Nessa fase do projeto, foi desenvolvido o serviço Web seguindo os requisitos definidos previamente e utilizando as tecnologias definidas para auxiliar o desenvolvimento desse projeto.

3.1.5 Realizar o estudo de caso

Nessa fase do projeto foi desenvolvida uma nova aplicação que utilize o serviço que foi desenvolvido, para que se fosse possível verificar o funcionamento do serviço, ou seja, se o mesmo atendia os requisitos definidos e assim validar se o objetivo do trabalho foi alcançado.

3.2 A SOLUÇÃO

A solução proposta por esse trabalho é um serviço Web que possibilite o seu usuário disponibilizar ou consumir os dados que esse serviço disponibiliza.

O serviço poderá ser utilizado de duas maneiras diferentes, ou seja, o serviço Web proposto por esse trabalho oferecerá ao seu usuário a possibilidade de disponibilizar qualquer dado e também permitirá que o usuário utilize os dados já existentes no serviço da maneira que desejar.

Para disponibilizar dados nesse serviço Web, é necessário que o cliente/usuário cumpra alguns pré-requisitos importantes, que são: ter uma base de dados pronta no formato CSV e com um modelo de banco de dados relacional definido para que possa ser feita a importação da base de dados para o banco de dados. Possuindo esses dois requisitos para fazer a disponibilização dos dados, agora é necessário fazer algumas adaptações no serviço para que sejam obtidas informações da base de dados do cliente e definir quais exemplos de consultas serão exibidos na documentação do serviço como exemplo para quem quiser utilizar os dados que vão ser disponibilizados através do serviço Web.

A outra forma de se utilizar o serviço é por meio da sua API. Nesse contexto de utilização é possível consumir os dados desse serviço de duas formas diferentes, que são: por meio de consultas feitas dinamicamente e através de uma aplicação cliente. Utilizando uma aplicação cliente desenvolvida pelo usuário interessado em consumir os dados desse serviço, a comunicação entre a aplicação cliente e a API do serviço será feita através de requisições REST por meio do protocolo HTTP feita da aplicação cliente para o serviço. Já para utilizar a API apenas para consultas o usuário terá a possibilidade de fazer consultas diretamente na URL da API, ou seja, o usuário poderá usar a URL base para esse tipo de consulta e poderá fazer suas consultas digitando os valores que deseja buscar diretamente na URL ou se preferir, poderá inserir esses valores em um formulário de consulta e submeter a mesma para obter o resultado que virá no formato JSON.

Nas sessões de 3.4 , 4.1 , 4.2 e 4.3 será explicado melhor como o serviço foi desenvolvido e os seus cenários de uso.

3.3 ARQUITETURA

Antes de definir as tecnologias que seriam utilizadas no serviço, foi necessário projetar como seria a arquitetura do mesmo, além de definir o que seria representado nessa arquitetura visando facilitar o entendimento do serviço de modo geral.

Com o objetivo de facilitar o entendimento do serviço de modo geral, foi elaborado um diagrama, como mostra a Figura 1, com uma visão “externa” do serviço, que vai desde a importação dos dados no formato CSV para o banco de dados relacional já modelado, até a comunicação de uma aplicação cliente com o servidor.

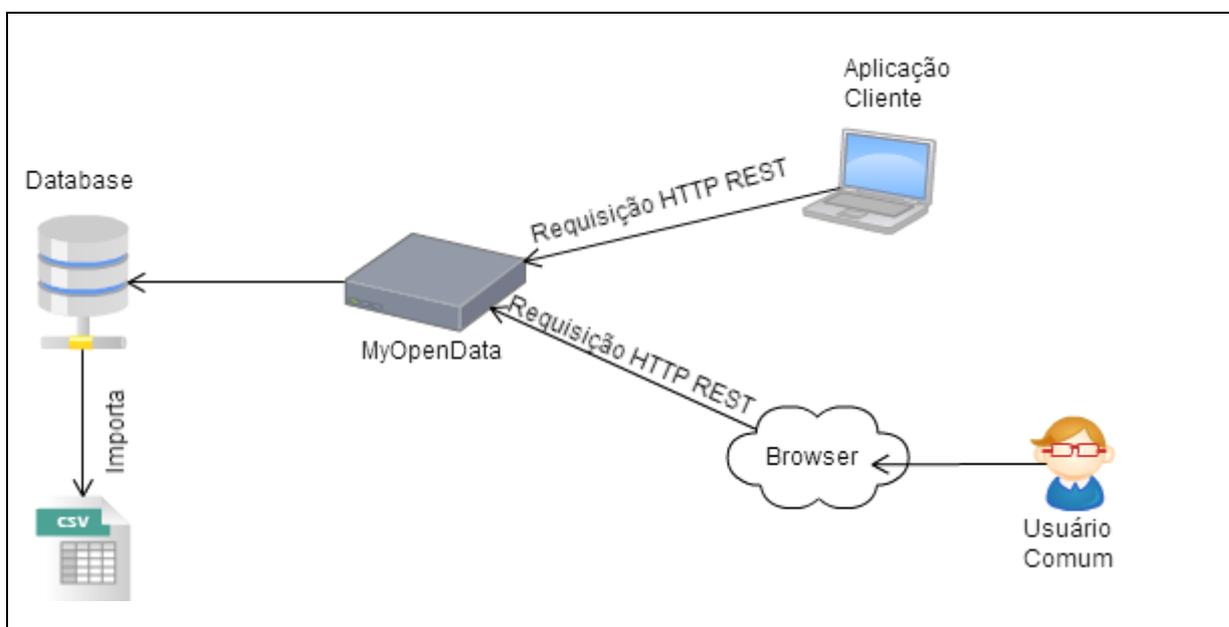


Figura 1 – Arquitetura geral do serviço

Conforme mostra a Figura 1, comunicação entre a aplicação cliente e o serviço é feita por meio de requisições HTTP REST, quando essa requisição é feita, um controller implementado dentro do serviço recebe essa requisição e repassa essa requisição para o DAO também implementado no serviço para que o mesmo faça a comunicação com o banco de dados e pegue as informações que a requisição está solicitando. Após o DAO recuperar as informações solicitadas pelo controller, será retornada uma lista para o controller com as informações solicitadas e o controller será responsável por fazer a serialização dessas informações para JSON e retornar para a aplicação cliente.

Além disso, também é mostrado na Figura 1, o padrão seguido pelo serviço em relação aos tipos de dados que o mesmo irá aceitar, no caso, a base de dados tem que está pronta e no

formato CSV para que esses dados possam ser importados para o banco de dados relacional previamente modelado.

Durante a criação desse diagrama, foi definido que os exemplos utilizados seriam genéricos, pois como se trata de um serviço que pode disponibilizar qualquer tipo de dado, não faz sentido modelar um diagrama utilizando um exemplo fixo, conforme é mostrado na Figura 2.

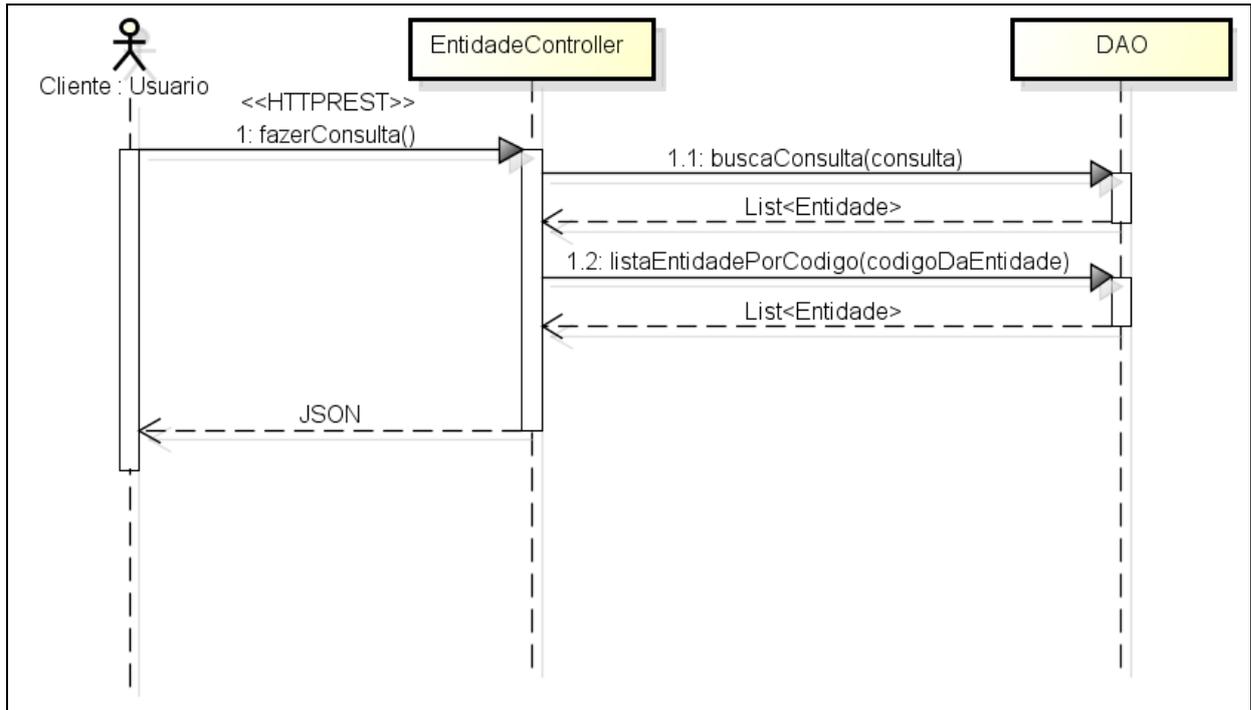


Figura 2 – Diagrama de Sequencia do Serviço

A Figura 2 mostra o diagrama de sequencia do serviço. Como se trata de um serviço que pode armazenar e disponibilizar qualquer tipo de dado, foi usado esse exemplo genérico para demonstrar o funcionamento interno do próprio serviço. Por definição e para facilitar uma possível manutenção no serviço, foi definido que cada entidade terá o seu controller e o seu DAO próprio, ou seja, caso alguma coisa tenha que ser mudado em alguma entidade, apenas os seus respectivos DAO e controller serão afetados.

3.4 IMPLEMENTAÇÃO E TECNOLOGIAS

Nessa sessão serão explicadas as tecnologias escolhidas para auxiliar o desenvolvimento do projeto, como também os motivos que levaram o autor a fazer suas escolhas. Além disso, também será explicado como a implementação do projeto foi feita.

3.4.1 DEFINIÇÃO DAS TECNOLOGIAS

Antes de iniciar o desenvolvimento do serviço, as tecnologias que seriam utilizadas no serviço foram definidas.

Como linguagem de programação utilizada no desenvolvimento do serviço, a linguagem escolhida foi Java, devido ao conhecimento que o autor possuía sobre essa linguagem e também por ser uma linguagem bastante difundida.

Em seguida, foi necessário definir qual tecnologia seria utilizada para apoiar o desenvolvimento desse serviço, e por se tratar de um serviço Web, as tecnologias escolhidas foram filtradas em serviços baseados no protocolo HTTP. As principais vantagens de utilizar tecnologias que se baseiam no protocolo HTTP, segundo Daigneau (2011), são, por exemplo, a facilidade de reutilização e compartilhamento de lógica comum com diversos clientes. Com a utilização do protocolo HTTP nesse trabalho, foi escolhido o REST pelo fato de ser leve e ter uma fácil manipulação. O REST é definido por Pautasso, et al (2008) como um estilo arquitetural para a construção de sistemas distribuídos e que possui quatro princípios básicos:

- Interações sem armazenamento de estado
- As mensagens são auto-descritas
- Interface padrão
- Recursos são identificados por uma única URI

O framework vRaptor 3 é um framework MVC para web com foco no desenvolvimento ágil que também foi escolhido para auxiliar o desenvolvimento do serviço. Alguns motivos levaram a essa escolha, como: Suportar a arquitetura REST, diminuir o tempo de trabalho com código repetitivo e também pelo conhecimento que o autor possui sobre o framework.

Devido as grandes vantagens que a Computação em Nuvem oferece para serviços Web, tanto em possibilitar um acesso mais fácil e rápido aos dados que serão disponibilizados, como também pelo custo acessível que essa tecnologia oferece, com isso foi definida a necessidade do serviço ser hospedado em uma nuvem a fim de explorar as vantagens que essa tecnologia oferece. Por limitações de tempo, foi utilizada a nuvem Amazon AWS apenas com escalabilidade vertical, ou seja, a capacidade das máquinas individuais pode ser ajustada caso o serviço esteja lento ou degradado.

3.4.2 IMPLEMENTAÇÃO DO SERVIÇO

Com as tecnologias definidas, surgiu a necessidade de definir como seriam estruturadas as URIs de acesso ao serviço. Levando em consideração ao estilo arquitetural adotado, no REST a realização de acesso aos recursos utiliza os métodos do protocolo HTTP que são POST, GET, PUT e DELETE, porém como o serviço não vai dar a opção dos seus usuários alterarem alguma informação no banco de dados, os métodos PUT e DELETE do protocolo HTTP não foram utilizados no desenvolvimento desse serviço.

Com a forma de como o acesso aos recursos do serviço ficou definida, e estando focada apenas nos métodos GET e POST, surgiu um novo requisito no sistema que permitiria ao usuário do serviço fazer suas próprias consultas sobre uma entidade disponível no serviço. Com isso o foi necessário fazer a implementação de um pequeno formulário para que o mesmo auxiliasse o usuário a fazer suas próprias consultas, conforme mostra a Figura 3.

Por meio desse formulário, o usuário poderá fazer suas consultas nas entidades do serviço, a Figura 3 mostra o formulário de consultas da entidade Escola e no formulário existe uma pequena documentação sobre os campos que o usuário pode utilizar para realizar sua consulta. Abaixo do formulário também tem uma pequena documentação sobre a resposta dessa consulta, contendo os campos que serão retornados na resposta, como também o tipo e sua descrição.

| Nome do Metodo / Entidade | | | | |
|--|----------------------|------------------------------|---|-----------------------------------|
| Parametros | | | | |
| Campo | Valor | Tipo | Obrigatorio | Descricao |
| Parametro 1 | <input type="text"/> | Tipo do Parâmetro | É obrigatorio para realizar a consulta? | Breve descrição sobre o parâmetro |
| Parametro 2 | <input type="text"/> | Tipo do Parâmetro | É obrigatorio para realizar a consulta? | Breve descrição sobre o parâmetro |
| <input type="button" value="Enviar Consulta"/> | | | | |
| Informações da Consulta | | | | |
| Será retornado um JSON com as seguintes informações | | | | |
| Campo | Tipo | Descricao | | |
| Parâmetro 1 | Tipo do Parâmetro | Descrição do valor retornado | | |
| Parâmetro 2 | Tipo do Parâmetro | Descrição do valor retornado | | |
| Parâmetro 3 | Tipo do Parâmetro | Descrição do valor retornado | | |
| Exemplo do JSON retornado | | | | |
| <pre>{ "list": [{ "parametro1": "valorDoParametro1", "parametro2": "valorDoParametro2", "parametro3": "valorDoParametro3" }] }</pre> | | | | |

Figura 3 – Tela de Consultas

Contudo, para dar mais uma opção de consulta para o usuário, o método POST não foi utilizado na implementação do serviço, ou seja, o serviço implementado nesse projeto trabalha apenas com requisições do tipo GET, essa escolha foi feita pensando na possibilidade do usuário também querer fazer suas consultas diretamente na URL.

Para auxiliar o usuário, algumas consultas foram previamente definidas, implementadas e documentadas para cada entidade do sistema seguindo os exemplos de consultas implementadas e documentadas no controller de cada entidade. Todos os controladores das entidades devem possuir basicamente a mesma estrutura, conforme mostra a Figura 4.

```
@Get
@Path(value="/nomeDaEntidade/{valorDoCodigoDaEntidade}")
public void listaEntidadePorCodigo(Long codigoDaEntidade){

    r.use(Results.json())
    .from(dao.listarEntidadePorCodigo(codigoDaEntidade))
    .serialize();
}
```

Figura 4 – Serialização das mensagens de resposta

Também foi necessário fazer a implementação de cada uma dessas consultas no DAO de cada entidade do serviço. Cada entidade segue a mesma estrutura que é mostrada na Figura

5, pois para facilitar o desenvolvimento é interessante usar os mesmos exemplos para cada entidade, mudando apenas alguns detalhes particulares de cada entidade quando essas mudanças forem necessárias para realizar essas consultas.

```
public Entidade listarEntidadePorCodigo(Long codigoDaEntidade) {  
    Entidade entidade = em.find(Entidade.class, codigoDaEntidade);  
    return entidade;  
}
```

Figura 5 – Requisição ao banco de dados com a consulta desejada

4 ESTUDO DE CASO

Nessa sessão serão apresentados os cenários de uso utilizados no estudo de caso, a base de dados escolhida pelo autor para realizar o estudo de caso e também o fluxo de execução do serviço.

4.1 CENARIO DE USO

Com as tecnologias definidas, foi necessário escolher uma base de dados para realizar o estudo de caso do serviço e para implementar a estrutura base do serviço utilizando o banco de dados escolhido. As bases de dados foram analisadas em relação a completude dos dados, a facilidade de se fazer uma modelagem SQL relacional e também a facilidade de normalizar a base de dados de acordo com as entidades escolhidas.

Após realizar esse estudo, foi selecionada para o estudo de caso a base de dados disponível em: <http://dados.gov.br/dataset/instituicoes-de-ensino-basico> fechando o escopo da base de dados para instituições de ensino básico do estado do Ceará e filtrando as informações que essa base de dados oferecia em apenas quatro entidades, foram elas: Município, Distrito, Escola e Recurso. Após definir as entidades que seriam utilizadas no sistema foi feita a importação dos dados do arquivo CSV disponibilizado pelo site citado anteriormente. A importação desses dados foi feita com a utilização da ferramenta pentaho que auxiliou nesse processo de importação de dados procurando diminuir ao máximo qualquer trabalho manual.

4.2 FLUXO DE EXECUÇÃO

O fluxo de execução do serviço é dividido em três partes: página inicial, documentação dos métodos e por fim página de consultas do serviço.

Na página inicial da aplicação, conforme mostra a Figura 6, é dada uma visão geral sobre como funciona o serviço e também é exibida uma pequena documentação sobre o serviço para que o usuário possa ter uma base de como o serviço funciona.

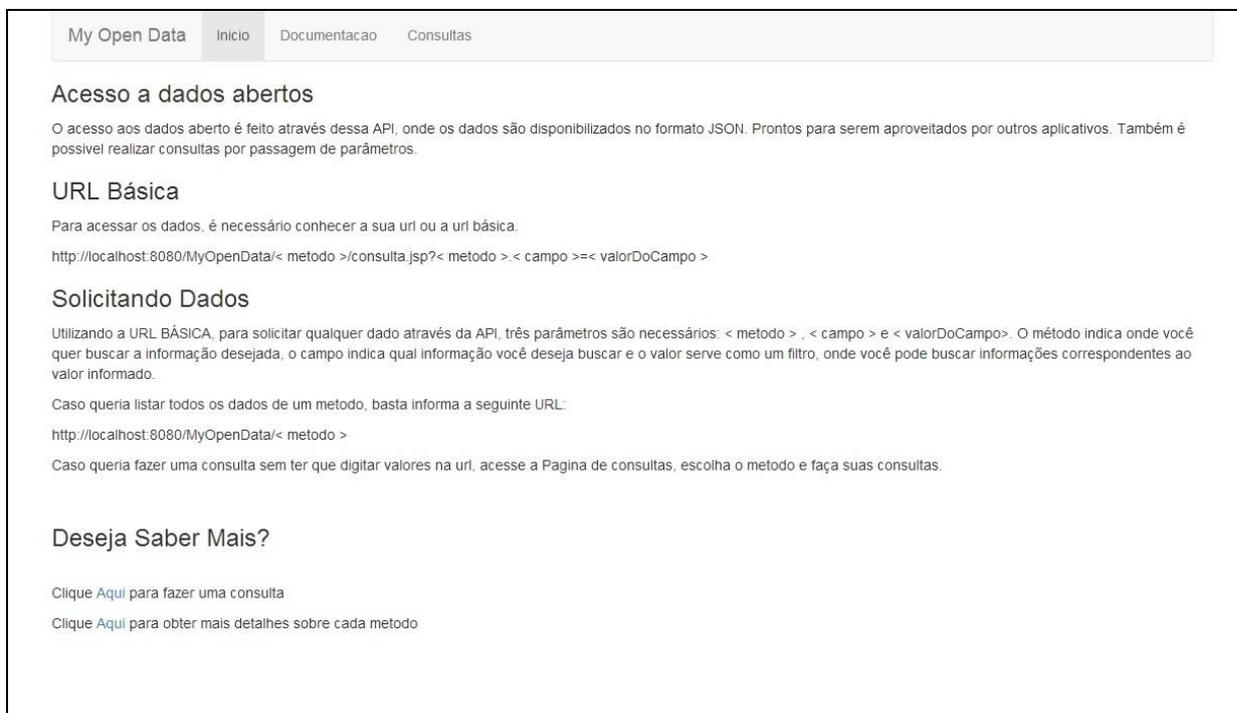


Figura 6 – Tela inicial

O serviço também possui uma pagina dedicada a documentação, nessa página é explicado com mais detalhes o funcionamento do serviço, quais os métodos disponíveis e também alguns exemplos são mostrados para facilitar o entendimento do usuário. A Figura 7, mostra a página de documentação apenas com os métodos disponíveis, foi adotado o estilo accordion do bootstrap para diminuir a quantidade de informações exibidas na tela do usuário, ou seja, o usuário só vai ver a documentação detalhada do método que ele quiser ver, para ver a documentação detalhada de um método, basta clicar no nome do método desejado, conforme mostra a Figura 8.

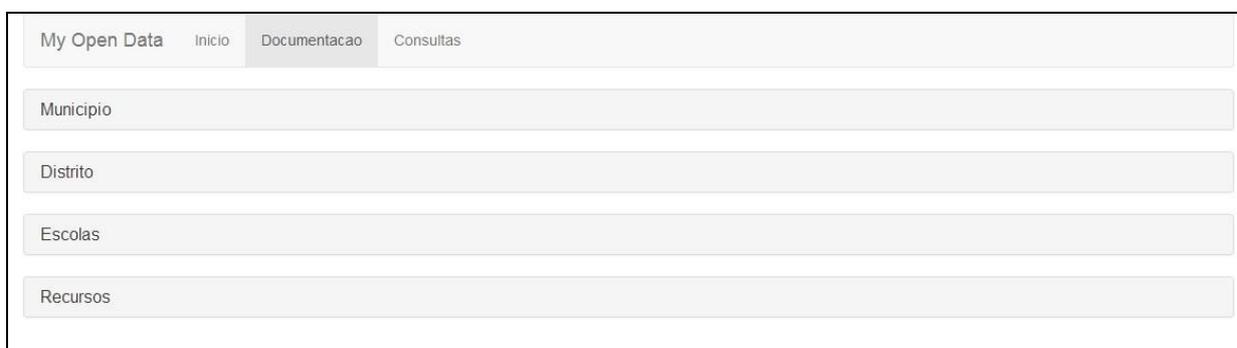


Figura 7 – Tela da Documentação

Distrito

Nome do metodo: distrito

Tipo de consulta: GET

Tipo de retorno: JSON

Alguns exemplos de utilização do metodo distrito:

Para listar um distrito buscando pelo seu codigo, basta acessar a URL

`http://< endereco >/distrito/codigoDistrito/< codigoDistrito >`

Exemplo - Consultar informações do distrito de São Miguel - Quixeramobim

URL: `http://localhost:8080/MyOpenData/distrito/codigoDistrito/294`

Nesse metodo será listado informações sobre o distrito de São Miguel

O retorno será em JSON

```
{
  "list": [
    {
      "codigoDistrito": 294,
      "endereco": "ASSENTAMENTO VISTA ALEGRE",
      "numero": "S/N",
      "zona": "Rural",
      "bairro": "",
      "nomeDistrito": "SAO MIGUEL",
      "cep": "63800000",
      "codigoMunicipio": 2311405
    }
  ]
}
```

Os campos retornados serão

- codigoDistrito
- bairro
- cep
- endereco
- nomeDistrito
- numero
- zona
- codigoMunicipio

Figura 8 – Documentação expandida do método distrito

O serviço também possui uma pagina para consultas que podem ser realizadas em cada método disponível pelo serviço, a página de consulta seguiu a mesma estrutura que a página de documentação, conforme mostra a Figura 9, para evitar que o usuário veja apenas informações que ele deseja.

My Open Data Inicio Documentacao **Consultas**

Municipio

Distrito

Escolas

Recursos

Figura 9 – Tela Consultas

As consultas podem ser feitas de duas formas, a primeira forma de consulta é feita por meio de um formulário, conforme mostra a Figura 10, esse modelo foi adotado para facilitar o usuário a estruturar sua consulta, mesmo que ele não possua conhecimento na linguagem SQL, além do formulário, também são exibidas informações sobre a resposta da consulta, no caso essas informações são sobre os campos que são retornados nas respostas e os seus respectivos tipos, além de uma pequena descrição. Outra forma de consulta é a passagem de parâmetros diretamente na URL padrão para esse tipo de consulta.

Escolas

Parametros

| Campo | Valor | Tipo | Obrigatorio | Descricao |
|-----------------|----------------------|--------|---|---|
| Codigo Escola | <input type="text"/> | Long | N | Codigo da Escola |
| Nome | <input type="text"/> | String | N | Nome da Escola |
| Rede | <input type="text"/> | String | N | Rede da Escola (Publica/Privada) |
| Situacao | <input type="text"/> | String | N | Situacao da Escola (Em Atividade/Extinta) |
| Codigo Distrito | <input type="text"/> | Long | S * Obrigatorio caso queira listar/buscar escolas de um distrito | Regiao do Municipio |

Informações da Consulta

Será retornado um JSON com as seguintes informações

| Campo | Tipo | Descricao |
|----------------|---------|--|
| codigoEscola | Long | Codigo da Escola |
| dDD | Integer | DDD do Telefone da Escola |
| dependencia | String | Dependencia da escola (Municipal/Estadual/Federal/Privada) |
| email | String | Email da Escola |
| nomeEscola | String | Nome da Escola |
| rede | String | Rede da Escola (Publica/Privada) |
| situacao | String | Situacao atual da Escola (Em Atividade/Extinta) |
| telefone | String | Telefone da Escola |
| codigoDistrito | Long | Codigo do distrito que a escola pertence |

Figura 10 – Tela consulta da entidade Escola

4.3 APLICAÇÃO CLIENTE

Foi desenvolvida uma aplicação cliente simples que usa todos os recursos disponibilizados pelo serviço desenvolvido nesse trabalho. Essa aplicação cliente foi desenvolvida em Java e tem como objetivo mostrar mais uma forma de utilização do serviço desenvolvido neste trabalho.

A aplicação cliente possui uma única funcionalidade, que é listar todos os municípios que possuem escolas que oferecem todos os recursos disponíveis na base de dados. Para isso foi necessário fazer uma consulta na aplicação via URL com todos os recursos disponíveis na base de dados com valor “Sim”, conforme mostra a Figura 11.

```

private static List<String> pegarCodigoDaEscola()
    throws MalformedURLException, IOException, JSONException {
    // build a URL
    String s = "http://localhost:8080/MyOpenData/recurso/consulta.jsp?recursos.biblioteca=Sim&"
        + "recursos.computador=Sim&recursos.internet=Sim&recursos.labCiencias=Sim&recursos.labInformatica=Sim";

    URL url = new URL(s);

    // read from the URL
    Scanner scan = new Scanner(url.openStream());
    String str = new String();
    while (scan.hasNext())
        str += scan.nextLine();
    scan.close();

    // build a JSON object
    JSONObject obj = new JSONObject(str);
    JSONArray a = obj.getJSONArray("list");

    List<String> escola = new ArrayList<String>();

    for(int i=0; i < a.length(); i++){
        JSONObject res = a.getJSONObject(i);
        String r = res.getString("codigoEscola");
        escola.add(r.toString());
    }

    return escola;
}

```

Figura 11 – Aplicação Cliente

Perceba que além da consulta via URL a no método *pegarCodigoDaEscola* também é feita a desserialização das mensagens de resposta dessa requisição que é feita no formato JSON.

Essa simples aplicação testa desde a possibilidade de se fazer uma consulta via URL até a utilização de alguns métodos já definidos na documentação do serviço.

Aplicação cliente faz uma requisição ao serviço por meio de uma consulta feita via URL, o serviço retorna a resposta da consulta no formato JSON. Após receber a resposta da requisição é feita a desserialização dessa resposta e é possível pegar valores necessários de acordo com sua necessidade, conforme foi mostrado na figura 10.

5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento de um serviço Web para a disponibilização e acesso a dados abertos procurando mostrar as vantagens e a importância desse tipo de serviço, além de abordar também as melhorias que a utilização da computação em nuvem pode impactar na utilização desse tipo de serviço.

Dessa forma, foram mostradas durante o desenvolvimento desse trabalho as dificuldades de se projetar um serviço para prover tanto a disponibilização de dados abertos como o acesso a esses dados e que as decisões que foram tomadas nesse trabalho buscaram um melhor aproveitamento das tecnologias utilizadas juntamente com o conceito de dados abertos no serviço.

Foi visto que a utilização do conceito de dados abertos está em crescimento desde a criação da Lei de Acesso a Informação e que a utilização da Computação em Nuvem juntamente com esse conceito de dados abertos aumenta a capacidade de utilização do serviço que poderá ser utilizado de qualquer lugar e por qualquer dispositivo com acesso a internet.

É importante destacar a dificuldade em projetar um serviço que possibilite a disponibilização de diferentes tipos de dados e que dependendo da base de dados escolhida, deve ser realizada uma adaptação no serviço para que o mesmo possa prover o acesso a esses dados da melhor maneira possível.

O próximo passo para esse trabalho é a possibilidade da migração do framework vRaptor 3 para a versão 4 e adicionar a possibilidade do usuário criar filtros para as consultas. Além disso, pretende-se implementar a característica de elasticidade da computação em nuvem e realizar experimentos que avaliem o desempenho da solução proposta.

É possível ainda desenvolver um framework ou um projeto pré-configurado que possa ser utilizado para o desenvolvimento de outras aplicações ou até mesmo para a disponibilização de bases de dados totalmente diferentes, diminuindo assim o tempo com a adaptação do serviço.

Por fim, conclui-se que não existe um padrão específico para a criação de serviços Web que possibilitam a disponibilização e acesso a dados abertos, possibilitando assim diferentes visões de como esse tipo de serviço pode funcionar e quais tecnologias se encaixam melhor para o desenvolvimento desse tipo de projeto, contudo todas essas visões de funcionamento e tecnologias tornam o desenvolvimento desse tipo de serviço algo desafiador.

REFERÊNCIAS

- ARANDA-ANDÚJAR, Andrés et al. AMADA: Web data repositories in the amazon cloud. In: **Proceedings of the 21st ACM international conference on Information and knowledge management**. ACM, 2012. p. 1-3.. Disponível em: < <http://hal.archives-ouvertes.fr/docs/00/73/06/87/PDF/top.pdf>>. Acesso em: 27 out. 2013.
- BREITMAN, Karin et al. Open government data in Brazil. **IEEE Intelligent Systems**, v. 27, n. 3, p.1-4, 2012. Disponível em: <http://www.inf.puc-rio.br/~casanova/Publications/Papers/2012-Papers/2012-IEEE-ISOGD_in_Brazil.pdf>. Acesso em: 27 out. 2013.
- BUYA, R. et al. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. **Future Gener. Comput. Syst.**, vol. 25, pp. 599–616, 2009.
- DADOS, Abertos. **Cartilha Técnica para Publicação de Dados Abertos no Brasil v1.0**. Disponível em: < <http://dados.gov.br/cartilha-publicacao-dados-abertos/> >. Acesso em: 21 out. 2013.
- DADOS, Abertos. **Guia de Abertura de Dados**. Disponível em: <<https://www.consultas.governoeletronico.gov.br/ConsultasPublicas/download.do?acao=arquivoDocumentoReferencia&tipo=pdf&id=93>>. Acesso em: 21 out. 2013.
- DADOS, Abertos. **O que são dados abertos?**. Disponível em: < <http://dados.gov.br/dados-abertos/> >. Acesso em: 21 out. 2013.
- DAIGNEAU, R.; **Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services**. 1 ed. Addison-Wesley Professional, 2011.
- DINIZ, V; Como conseguir dados governamentais abertos. In: **CONGRESSO CONSAD DE GESTÃO PÚBLICA, III, Brasília**. 2010. Disponível em: <http://www.escoladegoverno.pr.gov.br/arquivos/File/Material_%20CONSAD/paineis_III_congresso_consad/painel_13/como_conseguir_dados_governamentais_abertos.pdf>. Acesso em: 19 out. 2013.
- EAVES, D. The three Laws of Open Government Data. Setembro de 2009. **Tradução retirada de apresentação do Seminário da W3C - Dados Governamentais Abertos: Acesso a Informação. Brasília**. 2009. Disponível em: <www.governoeletronico.gov.br/anexos/apresentacao-w3c> Acesso: 21 out. 2013.
- ENDLER, M; VITERBO, J; FONSECA, H. Relatório 3 Volume Unico (2011). Disponível em: <<http://www-di.inf.puc-rio.br/~endler/paperlinks/TechReports/CPqD-rel3.pdf>>. Acesso em: 19 out. 2013.
- LEI GERAL DE ACESSO À INFORMAÇÃO PÚBLICA. Disponível em: <<http://legis.senado.gov.br/legislacao/ListaTextoIntegral.action?id=244430&norma=264259>> Acesso em: 25 out. 2013.

LOPES, G. A.; JUNIOR, J. M. EDUCLOUD 2: Implementando mecanismos de elasticidade em uma nuvem privada para ambientes acadêmicos. **Revista da Graduação**, Porto Alegre, v. 5, n. 2, 2012. Disponível em: <<http://revistaseletronicas.pucrs.br/ojs/index.php/graduacao/article/view/12420/8338>>. Acesso em: 19 out. 2013.

OPEN GOV DATA. **Eight principles of open government data**. Dezembro de 2007. Disponível em: <<http://www.opengovdata.org/home/8principles>> Acesso: 21 out. 2013.

PAUTASSO, C.; ZIMMERMANN, O.; LEYMANN, F.; **Restful Web Services vs. “Big” web services**. Proceeding of the 17th international conference on World Wide Web - WWW '08. New York, USA: ACM Press, 2008.

RODRIGUES, F. A. ; SANTANA, R. C. G. . Padrão para dados abertos sobre o transporte público rodoviário urbano. 2010. **VI Encontro Internacional de Informação, Conhecimento e Ação**, pag 1-8, 2010.

SOUSA, F. R. C. et al. Gerenciamento de dados em nuvem: Conceitos, sistemas e desafios. In **SBBB**, pag. 101–130, 2010. Disponível em: <<http://www.es.ufc.br/~flavio/papers/sbbd2010.pdf>> Acesso: 21 out. 2013.

SOUSA, F. R. C., et al. Quality of service for database in the cloud. In **2rd International Conference on Cloud Computing and Services Science (CLOSER) '12**, pp. 595–601, 2012.

SOUSA, F. R. C.; MOREIRA, L. O.; MACHADO, J. C. Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios. In: III Escola Regional de Computação Ceará – Maranhão – Piauí (ERCERMAPI), 2009, Piauí. **Anais do ERCERMAPI**. Piauí: ERCERMAPI, 2009. Disponível em: <<http://www.es.ufc.br/~flavio/papers/ercemapi2009.pdf>>. Acesso em: 19 out. 2013.

SCHAD, J.; DITTRICH, J.; QUIANÉ-RUIZ, J.-A. Runtime measurements in the cloud: Observing, analyzing, and reducing variance. **PVLDB**, vol. 3, pp. 460–471, 2010. Disponível em: <http://delivery.acm.org/10.1145/1930000/1920902/p460schad.pdf?ip=187.19.243.165&id=1920902&acc=ACTIVE%20SERVICE&key=C2716FEBFA981EF1AF9B91FE89BF60B67ECB63BBA3A414A2&CFID=267903659&CFTOKEN=80706209&__acm__=1386202127_43db952a52e82656f396370884bfe2b2>. Acesso em: 19 out. 2013.

TAURION, Cezar. **Cloud Computing: Computação em Nuvem Transformando o Mundo da Tecnologia da Informação**. Rio de Janeiro: Brasport, 2009.

TRINDADE, L. N. et al. Legibilidade em Dados Abertos: uma experiência com os dados da Câmara Municipal de São Paulo. 2012. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/sbsi/2012/0068.pdf>>. Acesso em: 19 out. 2013.

ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. **Journal of Internet Services and Applications**, pp. 7–18, 2010. Disponível em: <<http://link.springer.com/content/pdf/10.1007%2Fs13174-010-0007-6.pdf>>. Acesso em: 19 out. 2013.