UNIVERSIDADE FEDERAL DA PARAÍBA

CENTRO DE CIÊNCIAS APLICADAS A EDUCAÇÃO

DEPARTAMENTO DE CIÊNCIAS EXATAS

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

UM SERVIÇO DE GESTÃO DE DICIONÁRIO PARA A SUÍTE VLIBRAS

Renan Soares de Andrade

Orientador: Prof. Dr. Tiago Maritan Ugulino de Araújo

RIO TINTO - PB

2015

Renan Soares de Andrade

UM SERVIÇO DE GESTÃO DE DICIONÁRIO PARA A SUÍTE VLIBRAS

Monografia apresentada para obtenção do título de Bacharel à banca examinadora no Curso de Bacharelado em Sistemas de Informação do Centro de Ciências Aplicadas e Educação (CCAE), Campus IV da Universidade Federal da Paraíba.
Orientador: Prof. Dr. Tiago Maritan Ugulino de Araújo.

RIO TINTO - PB
2015

Ficha catalográfica preparada pela Seção de Catalogação e Classificação da Biblioteca da UFPB

A553s Andrade, Renan Soares de.
Um serviço de gestão de dicionário para a suíte VLIBRAS. / Renan Soares de Andrade. – Rio Tinto: [s.n.], 2015.
58 f. : il.-

Orientador (a): Prof. Dr. Tiago Maritan Ugulino Araújo.
Monografia (Graduação) – UFPB/CCAE.

1. Software - desenvolvimento. 2. Libras - tradutor. 3. Língua Brasileira de Sinais.

UFPB/BS-CCAE *CDU: 004.41 (043.2)*

Renan Soares de Andrade

UM SERVIÇO DE GESTÃO DE DICIONÁRIO PARA A SUÍTE VLIBRAS

Trabalho de Conclusão de Curso submetido ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal da Paraíba, Campus IV, como parte dos requisitos necessários para obtenção do grau de BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Assinatura do autor:_____

APROVADO POR:

_____

Orientador: Prof. Dr. Tiago Maritan Ugulino de Araújo
Universidade Federal da Paraíba – Campus V

_____

Msc. Danilo Assis Nobre dos S. Silva
Laboratório de Aplicações de Vídeo Digital

_____

Prof. Dr. Carlos Eduardo Silveira Dias
Universidade Federal da Paraíba – Campus IV

RIO TINTO - PB
2015

A Deus, pelo dom da vida, por todas as conquistas e pela saúde para chegar até aqui.

AGRADECIMENTOS

Agradeço primeiramente a Deus pelo dom da vida, por me permitir e me ajudar a chegar até aqui, por não me abandonar e por tudo me ajudar em tudo que conquistei até hoje.

Aos meus pais, minha mãe Valdenez e ao meu pai Geraldo, por toda a educação, paciência, ajuda, confiança e pela força em todos os momentos da minha vida, dedico todas minhas conquistas a vocês. Aos meus irmãos, Gernan e Natalhia pelo apoio, paciência e amor de sempre. Sou muito grato a vocês e agradeço a Deus todos os dias pela família que tenho.

Aos meus amigos Ana Paula, Larissa, Luis Feliphe, Edivaldo, Joeliton, Williby, Cryspo, Neto, Chianca, Lais, Galdino, João Joaquim Neto, Hyago (*in memoriam*), Janielly, Emilayne, Jackson, Tatiana que a vida me presenteou a partir da universidade, vocês foram muito importantes nessa caminhada, os diversos momentos compartilhados foram essenciais para conseguir chegar ao fim desta caminhada e obter o êxito desejado.

Aos meus amigos Lucas, Tati, Michel, Andressa, Ana Rosa, Jobyson, Sávio, Antônio, João, Cayo, Paloma, Guilherme que a vida me presentou aos poucos no dia-a-dia ao longo destes 23 anos de caminhada, guardo cada um de vocês no meu coração pelos diversos momentos, pela força de sempre, pelo braço amigo.

A todos os professores que estiveram comigo nesta caminhada, de modo especial aos professores do Departamento de Ciências Exatas em nome do professor Rafael Magalhães (coordenador do curso) e ao professor Rodrigo Rebouças (coordenador do departamento).

A todos meus companheiros que estiveram comigo no Laboratório de Aplicações de Vídeo Digital (LAViD), Leonardo, Erickson, Icaro, Jaelson, Danilo, Manuella, Samara, André, Felipe, Vandhuy, Felipe Lacet, Yúrika, Victor, a qual tenho o privilegio de compartilhar diariamente diversos momentos de aprendizagem, trabalho e descontração. Ao meu orientador Tiago Maritan pela sua disposição e ajuda de sempre, por todo o conhecimento repassado e pelo exemplo de pessoa.

RESUMO

Os surdos enfrentam diversas barreiras para ter acesso à informação. A inclusão deles, no entanto, perpassa pelo reconhecimento da sua língua natural que é a língua de sinais, posto que, as línguas orais são consideradas sua segunda língua. Em virtude disso, eles possuem dificuldade de ler e escrever na língua oral do seu país, o que acaba criando uma série de barreiras de acesso à informação e à comunicação. Somado a isso, as Tecnologias de Informação e Comunicação (TICs), quando projetadas, raramente levam em consideração as necessidades específicas desses usuários. Na TV, por exemplo, o suporte a línguas de sinais é geralmente limitado a uma janela com intérprete humano, que possui, em geral, um alto custo operacional. Além disso, a tradução manual para língua de sinais, geralmente, não é eficaz em ambientes como a Web, que possuem conteúdos dinâmicos, que mudam a todo instante. Para endereçar esses problemas, existem no mercado algumas ferramentas de tradução automática, mas elas, em geral, são soluções fechadas e não possuem APIs públicas para integração com outros serviços. O objetivo deste trabalho é desenvolver um serviço de dicionário chamado DaaS (*Dictionary as a Service*), que será um serviço de provimento, controle e gerenciamento dos sinais da suíte de ferramentas de tradução automática VLIBRAS, tornando-o um dicionário aberto de Língua Brasileira de Sinais (LIBRAS). Como isso, pretende-se facilitar a criação de serviços que envolvam dicionários de LIBRAS, integração de ferramentas de tradução automática para LIBRAS, dentre outros. Através desse serviço, os usuários poderão montar um dicionário de LIBRAS próprio a partir de vídeos, animações geradas com um avatar-3D ou metadados correspondente aos sinais; baixar um grupo de sinais desse Dicionário; fazer controle de versões, cópias, consultas desse dicionário, dentre outros. Para desenvolver esse serviço fez-se inicialmente, uma análise das necessidades dos possíveis usuários; a documentação dos requisitos funcionais e não-funcionais; e a elaboração de uma arquitetura a partir dos requisitos levantados. Assim, desenvolveu-se o serviço com base nos requisitos e na arquitetura proposta utilizando a linguagem de programação JavaScript com as tecnologias Node.js e MongoDB. A integração do DaaS com o VLIBRAS foi validada através de um pré-teste utilizando o controle de atualizações do dicionário de sinais das aplicações da suíte VLibras.

Palavras-chave: acessibilidade, língua brasileira de sinais, dicionário de sinais, vlibras.

ABSTRACT

Deaf people face many barriers to be granted access to information. Their inclusion, however, is permeated by the knowledge that their first language is the sign language, given that oral dialects are considered their second language. Because of this, they might have difficulties reading and writing in their country's official language, creating a series of information access and communication problems. In addition to this, information and communication technologies (ICTs) hardly ever take these users specific needs into account when designed. For instance, TV support for sign language is usually limited to a small window on the corner of the screen with a human interpreter, generating high operational costs. Besides, manual translation to sign language is not usually effective on environments such as the Web, with dynamic, rapidly changing content. To address these issues, there are a few automatic translation tools available on the market, but they are generally closed and don't offer public APIs for integration with other services. This objective of this work is to develop a service called DaaS (Dictionary as a Service) that will provide control and management of the signs in the automatic translation tool suite VLIBRAS, turning it into an open dictionary for the Brazilian Sign Language (LIBRAS). Thus, it is intended to facilitate the creation of services containing dictionaries of LIBRAS, the integration of automatic translation tools to LIBRAS, among others. Using this service, users will be able to create their own LIBRAS dictionary through videos, animations with a 3D avatar or metadata corresponding to the signs; make version control, copies and searches on this dictionary, among others. To develop this service, the necessities of the users needs were analyzed, functional and non-functional requirements documented and an architecture based on these requirements was created. The service based on these architecture and requirements was developed using the JavaScript programming language with Node.js and MongoDB. DaaS integration with VLIBRAS was validated through a pretest using the update control of the VLibras applications sign languages dictionary.

Keywords: accessibility, brazilian sign language, dictionary of signals, vlibras.

LISTA DE FIGURAS

Figura 1: Componentes da VLibras-Box 24
Figura 2: Modelo de conexão web multi-thread 27
Figura 3: Modelo de conexões web single-thread 28
Figura 4: Ferramenta do Handtalk para tornar sites acessíveis....33
Figura 5: Arquitetura do DaaS....41
Figura 6: Representação da Integração do DaaS com a Suíte VLIBRAS 45
Figura 7: Exemplo de requisição com retorno no formato JSON utilizando o POSTMAN....46
Figura 8: Exemplo de requisição com retorno no formato XML utilizando o POSTMAN 47
Figura 9:Tela de atualização do dicionário do VLibras-Desktop Windows....48
Figura 10: Tela principal do VLibras-Desktop Windows com opções de configuração....48
Figura 11: Tela principal do VLibras-Desktop informando sobre atualização....49
Figura 12: Diagrama com exemplos de requisições a funcionalidade de baixar sinais.....51
Figura 13: Diagrama com exemplos de requisições a funcionalidade de consultar versão do dicionário. 52

LISTA DE TABELAS

Tabela 1: Relação de ferramentas e informações sobre os seus respectivos dicionário de sinais 34
Tabela 2: Requisito funcional “Listar sinais” do DaaS 36
Tabela 3: Requisito funcional "Consultar sinal" do DaaS 36
Tabela 4: Requisito funcional "Consultar versão do Dicionário" do DaaS 37
Tabela 5: Requisito funcional “Baixar sinais” do DaaS 37
Tabela 6: Requisito funcional "Listar Requisições a sinais" do DaaS....38
Tabela 7: Requisito funcional "Cadastrar sinal" do DaaS 38
Tabela 8: Requisito não-funcional "Restrição para consultas" do DaaS 39
Tabela 9: Requisito não-funcional "Restrição para cadastro de sinais" do DaaS....40
Tabela 10: Requisito não-funcional "Restrição para baixar sinais"....40
Tabela 11: Informações do teste de carga e desempenho da funcionalidade de sinais sob demanda 50
Tabela 12: Informações do teste de carga e desempenho da funcionalidade de consultar versão do dicionário. 52

LISTA DE SIGLAS

JSON	JavaScript Object Notation
XML	Extensible Markup Language
LIBRAS	Língua Brasileira de Sinais
SIG	Sistemas de Informação Gerenciais
LAVID	Laboratório de Aplicações em Vídeo Digital
BD	Banco de Dados
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
API	Application Programming Interface
DAAS	Dictionary As a Service

SUMÁRIO

RESUMO ..... **VIII**

ABSTRACT ..... **IX**

LISTA DE FIGURAS ..... **X**

LISTA DE TABELAS ..... **XI**

LISTA DE SIGLAS ..... **XII**

1 INTRODUÇÃO **15**

1.1 OBJETIVOS 17

1.1.1 Objetivos gerais ***17***

1.1.2 Objetivos específicos ***17***

1.2 METODOLOGIA 18

1.3 ESTRUTURA DA MONOGRAFIA 19

2 FUNDAMENTAÇÃO TEÓRICA **20**

2.1 ACESSIBILIDADE 20

2.2 LÍNGUAS DE SINAIS 21

2.2.1 LIBRAS ***22***

2.3 TRADUÇÃO AUTOMÁTICA 22

2.3.1 Suíte VLIBRAS ***23***

2.4 QD FRAMEWORK 25

2.5 SERVIÇOS WEB 26

2.5.1 Interface de Programação Aplicada ***26***

2.5.2 Node.js ***26***

2.5.3 MongoDB ***29***

3 TRABALHOS RELACIONADOS **30**

3.1 DICIONÁRIO ACESSO BRASIL 30

3.2 RYBENA 30

3.3 POLI-LIBRAS 31

3.4 TLIBRAS 31

3.5 FALIBRAS 32

3.6 PRODEAF 32

3.7 HANDTALK 33

3.8 QUADRO COMPARATIVO 34

4 SOLUÇÃO PROPOSTA **35**

4.1 REQUISITOS 35

4.1.1 Requisitos Funcionais ***36***

4.1.2 Requisitos Não-Funcionais ***39***

4.2 ARQUITETURA DO SERVIÇO 40

4.3 DICTIONARY AS A SERVICE 42

4.4 TECNOLOGIAS UTILIZADAS 44

5 RESULTADOS E DISCUSSÕES **45**

5.1 INTEGRAÇÃO COM O VLIBRAS 45

5.2 TESTES COMPUTACIONAIS 49

5.2.1 Testes com sinais específicos sob demanda 50

5.2.2 Consulta de versão do dicionário 51

6 CONSIDERAÇÕES FINAIS 54

6.1 SUGESTÕES DE TRABALHOS FUTUROS 54

REFERÊNCIAS BIBLIOGRÁFICAS 56

1 INTRODUÇÃO

Na sociedade moderna, existem diversas formas de se definir o termo acessibilidade. A mais comum está relacionada ao compromisso de melhorar a qualidade de vida das pessoas, principalmente com relação aos fatores perceptivos, cognitivos e motores (Gallahue e Ozmun, 2005). Com isso, torna-se necessário prover a acessibilidade para reduzir barreiras de acesso à informação e comunicação dos deficientes auditivos.

A promoção da acessibilidade dos deficientes auditivos perpassa pela necessidade de comunicação. Segundo Russell e Norvig (2004), "a comunicação é a troca intencional de informações provocada pela produção e percepção de sinais extraídos de um sistema compartilhado de sinais convencionais". Através desses sistemas compartilhados de sinais, denominados línguas, é que se torna possível a transmissão de informações entre indivíduos.

A comunicação dos ouvintes é realizada por intermédio de línguas orais, ou seja, através de sons articulados que são percebidos pelo sistema auditivo. Os surdos, por outro lado, comunicam-se através de línguas gestuais, denominadas línguas de sinais. Essas línguas, diferentemente das línguas orais que utilizam o som na comunicação, utilizam um canal visual, isto é, um conjunto de elementos linguísticos manuais, corporais e faciais para articular os sinais. O emissor constrói uma sentença a partir desses elementos linguísticos e o receptor utiliza o sistema visual para compreender o que está sendo comunicado. Isso implica que para as pessoas surdas, as línguas orais, utilizadas cotidianamente pela maioria das pessoas e em praticamente todos os meios de comunicação, quando conhecidas, representam apenas "uma segunda língua".

Além disso, uma vez que as línguas possuem uma grafia baseada em sons, muitos deles passam vários anos na escola e não conseguem aprender a ler e escrever na língua oral de seu país. No Brasil, por exemplo, segundo o censo demográfico do IBGE de 2000 [IBGE 2000], cerca de 97% das pessoas surdas não concluem o ensino médio. Um estudo com crianças e adolescentes surdos holandeses de 7 a 20 anos de idade, mostrou que apenas 25% deles possuem uma capacidade de leitura igual ou superior ao de uma criança sem deficiência de 9 anos.

Embora as línguas de sinais sejam reconhecidas como língua natural, a comunidade surda de todo o mundo ainda tem dificuldade em ter acesso à informação usando estas línguas, o que limita o surdo a ter uma vida em sociedade, dificultando a comunicação e acesso a informações.

No mundo existem diversas línguas de sinais, cada uma contendo suas próprias regras gramaticais, vocabulários e fonemas (BUTTUSSI; CHITTARO; COPPO, 2007). Como exemplo, podem ser citadas a língua americana de sinais (American Sign Language - ASL)

(STOKOE, 1980) (MEIRELLES; GALVÃO, 2004), a língua britânica de sinais (British Sign Language - BSL) (STOKOE, 1980), a língua de sinais espanhola (Spanish Sign Language - SSL) (SAN-SEGUNDO et al., 2006) e a língua francesa de sinais (French Sign Language - FSL) (STOKOE, 1980). No Brasil a Língua Brasileira de Sinais (LIBRAS) é a utilizada pela maioria dos surdos brasileiros e foi reconhecida pela Lei n 10.436, de 24 de abril de 2002.

Na medida em que a LIBRAS foi reconhecida e disseminada, as tecnologias também vêm evoluindo e contribuindo para a comunicação dos surdos por meio da língua de sinais. Como um exemplo desse tipo de tecnologia, pode-se destacar o grande uso de vídeos por usuários surdos em plataformas como o YouTube.

Para Silva (2010):

> "(...) *é preciso estar a par da novidade digital que permite autonomia, por colaboração na manipulação das informações que ganham sentido por meio das ações de cada indivíduo que deixa de ser mero receptor para tornar-se também emissor de informações*" (SILVA, 2010, p. 137)

A disseminação das tecnologias e da LIBRAS estão colaborando no convívio dos surdos e na sua inclusão na sociedade. Porém, a maioria dos sítios e aplicativos ainda apresentam conteúdos ou em língua portuguesa ou outras línguas orais.

Uma das soluções para a inclusão da LIBRAS nessas diversos tecnologias é a tradução manual da língua portuguesa para a língua de sinais. No entanto, essa modalidade de tradução possui alguns problemas que devem ser levados em consideração, dentre eles:

- Alto custo de produção: Segundo o Sindicato Nacional de Tradutores (SINTRA), a tradução de um áudio em língua portuguesa de 60 minutos para LIBRAS custa R$ 585,00 e a tradução de cada em *Sign Writing*[1] custa R$ 0,32.
- Dinamismo dos conteúdos na era da informação: as informações estão em constante mudança e todos os dias novos sítios e aplicações são criados e atualizados.

Diante desses problemas, uma das alternativas seria então a utilização de ferramentas de tradução automática para línguas de sinais. Essas ferramentas, em geral, dependem de uma base de dados (dicionário de sinais), que é composta por milhares de vídeo, animações ou metadados para sinalizar as informações em língua de sinais.

[1] O *Sign Writing* é um sistema de escrita utilizado para escrever as línguas de sinais, expressando a configuração das mãos, os movimentos, as expressões faciais e os pontos de articulação.

A Suíte VLIBRAS é um conjunto de ferramentas computacionais de código aberto, responsável por traduzir automaticamente conteúdos digitais (texto, áudio e vídeo) em Língua Brasileira de Sinais (LIBRAS), tornando computadores, dispositivos móveis e plataformas Web acessíveis para pessoas surdas. Ela é composta pelas ferramentas VLibras-Plugin, VLibras-Desktop, VLibras-Video e WikiLibras.

Para contribuir no desenvolvimento desses tradutores, o objetivo deste trabalho é propor a criação do DaaS (*Dictionary as a Service*), um serviço para gestão e provimento de um dicionário de LIBRAS aberto com controle de versões e integração com a Suíte VLIBRAS.

A partir do DaaS pretende-se criar uma solução para suprir toda a demanda de sinais para todas as ferramentas da suíte VLIBRAS e oferecer um *webservice* API para que os usuários terceiros possam desenvolver suas próprias aplicações utilizando os sinais, disponibilizado em vários formatos. Com o serviço, pretende-se também estimular e facilitar a criação de novas aplicações que usam tradução automática, em especial aquela que venham a reduzir barreiras na comunicação e na inclusão dos surdos.

1.1 OBJETIVOS

1.1.1 Objetivos gerais

O objetivo geral do trabalho é desenvolver um serviço para gestão e provimento de um dicionário de LIBRAS aberto (DaaS), o qual conterá uma API pública para integração com outras aplicações e com controle de versões. Desse modo, pretende-se criar uma solução para suprir a demanda de sinais para todas as ferramentas da suíte VLIBRAS e prover uma API *webservice* para facilitar o desenvolvimento de aplicações que possam fazer a tradução automática para LIBRAS, independentemente da plataforma ou do tipo de ferramenta em que se pretenda desenvolver, tais como TV, Web, Cinema ou Desktop.

1.1.2 Objetivos específicos

Para alcançar o objetivo geral deste trabalho, os seguintes objetivos específicos foram definidos:

- Levantar os requisitos funcionais e não-funcionais do serviço com usuários surdos e desenvolvedores de serviços de tradução automática para Libras;
- Projetar uma arquitetura do serviço a partir da análise destes requisitos**;**

- Desenvolver o serviço de dicionário para LIBRAS e validá-lo na concepção de outros serviços.
- Integrar o serviço com algumas aplicações da plataforma Suíte VLIBRAS para validar uma parte da solução proposta.

1.2 Metodologia

Para desenvolvimento deste trabalho, foram planejadas e realizadas as seguintes atividades:

Atividade 1 - Análise bibliográfica: levantamento bibliográfico dos principais trabalhos relacionados à tradução automática de Português para a LIBRAS. Pesquisar se existe e como funcionam os dicionários da língua de sinais desses trabalhos.

Atividade 2 - Análise de necessidades e levantamento dos requisitos: análise das necessidades dos prováveis usuários das funcionalidades, simulando o seu uso e analisando quais seriam as melhores maneiras de prover os sinais e quais seriam os formatos mais adequados para cada aplicação e suas devidas tecnologias. Após a análise das necessidades do usuário em relação ao serviço, foi realizada a elaboração dos requisitos funcionais e não-funcionais com a descrição baixo nível das funcionalidades do sistema a partir da constatação de elementos que supram a demanda de uso do serviço.

Atividade 3 - Definição de arquitetura: definição da arquitetura baixo nível do sistema pensando em termos de hardware, quantos servidores seriam necessário para suprir a demanda e como iria ser a comunicação entre os mesmos, suas localizações e necessidades de configuração.

Atividade 4 - Análise de tecnologias: busca das tecnologias e elaboração de uma análise comparativa das tecnologias disponíveis do mercado que se atendessem os requisitos definidos, de acordo com a melhor performance da tecnologia.

Atividade 5 - Desenvolvimento do serviço: A partir da arquitetura e dos requisitos documentados das tecnologias levantadas anteriormente, o sistema foi desenvolvido utilizando o Node.js e integrado ao MongoDB como banco de dados.

Atividade 6 – Validação do serviço: realizar um conjunto de experimentos do serviço, integrando a proposta a uma ferramenta de tradução automática.

1.3 Estrutura da Monografia

A estrutura está organizada da seguinte forma: no primeiro capítulo foi apresentada uma contextualização do problema de pesquisa. O segundo capítulo apresenta uma fundamentação teórica dos assuntos inerentes à linha de pesquisa desse trabalho. Nele, alguns conceitos chave como acessibilidade, línguas de sinais, Língua Brasileira de Sinais, tradutores automáticos e conceitos das tecnologias envolvidas na solução proposta são apresentados com mais detalhes.

O terceiro capítulo apresenta alguns trabalhos relacionados com a problemática que este trabalho busca solucionar. Os trabalhos foram selecionados de acordo com o grau de relevância com a solução proposta.

O quarto capítulo apresenta as características inerentes ao desenvolvimento da solução proposta neste trabalho. Neste capítulo são expostos os requisitos funcionais e não-funcionais, a arquitetura da solução, como as tecnologias citadas anteriormente foram utilizadas para o desenvolvimento da solução e uma arquitetura de integração. Além de apresentar uma integração da solução proposta, explicando como em detalhes a integração foi realizada e os testes automáticos e manuais para validar a solução proposta.

O quinto capítulo irá apresentar os primeiros resultados a partir da solução proposta, em que se apresentam e debatem os resultados reais do serviço proposto.

Por fim, o sétimo capítulo expõe as considerações finais, discutindo os aspectos gerais em relação aos objetivos definidos. São sugeridos alguns trabalhos futuros, a fim de melhorar e validar a solução proposta.

2 FUNDAMENTAÇÃO TEÓRICA

O aporte teórico do presente trabalho será apresentado neste capítulo que inicia expondo as características sobre acessibilidade e Língua de Sinais, enfatizando a Língua Brasileira de Sinais (Libras). Em seguida, aborda-se a tradução automática destacando os tradutores automáticos de Português para LIBRAS.

2.1 Acessibilidade

Tornar algo acessível referenda-se como ato grandioso. Permitir que se estabeleça uma ponte entre um desejo e uma realidade vem sendo o objetivo que torna aquela robusta expressão um mecanismo da diminuição de distâncias. A sociedade mundial, a partir da construção de instrumentos que possibilitam a redução das diferenças, observando as peculiaridades de cada uma delas, permite que elas não afastem os sonhos dos indivíduos ao acesso às mais variadas realidades. Assim, a preocupação com a acessibilidade não é apenas uma contemplação de regras internacionais ou mera observância de diferentes normas, mas um avanço na construção de caminhos que permitem o acesso a lugares que antes não podiam ser visitados, revelando assim a concretização do mais puro sinônimo de humanidade.

Diante disso e das revoluções empreendidas no século passado, o acesso à informação passou a ser inegavelmente mais dinâmico. Contudo, tal velocidade precisa também ser contemplada pelo maior número de pessoas e de acordo com as suas necessidades. As tecnologias devem, portanto, funcionar como aliadas na redução das desigualdades e não servir como um mecanismo ainda mais desagregador. Dessa forma, a acessibilidade digital revela a possibilidade de que pessoas com deficiência possam interagir integralmente, por meio de instrumentos redutores de desigualdades, compreendendo as informações ali dispostas e se utilizando dos meios tecnológicos disponíveis para compreensão e construção de novas realidades.

Dentro deste grupo de pessoas com necessidades especiais, este trabalho debruçou-se sobre as pessoas surdas e com redução na capacidade auditiva, objetivando analisar as dificuldades encontradas por aquelas e buscando alternativas para facilitar a inclusão delas no mundo digital e, consequentemente, na sociedade.

Os surdos encontram barreiras substanciais quanto a leitura e escrita com a língua portuguesa. Para facilitar neste processo de inclusão, atualmente existem tecnologias

desenvolvidas que permitem o acesso a comunicação. Contudo, a capacitação de profissionais e a garantia de pleno acesso a todas as pessoas que deveriam ser contempladas com aqueles instrumentos representam alguns entraves a serem enfrentados pelo desenvolvimento das TICs.

2.2 Línguas de Sinais

A comunicação entre os surdos é realizada por meio de línguas gestuais, chamadas línguas de sinais, que é a modalidade de comunicação natural dos surdos. Essas não se realizam pelo canal oral-aditivo e sim pelo campo visual e espacial.

Normalmente, algumas pessoas possuem uma visão equivocada sobre as línguas de sinais, considerando-as como uma interpretação das línguas orais. No entanto, ela são línguas próprias e contêm toda uma estrutura linguística específica, assim como as línguas orais. [Brito 1995] [Quadros 1997]. Elas são línguas de natureza gestual-visual porque se realizam pelo canal visual com gestos, são independentes das línguas orais e também são consideradas línguas porque possuem diversos níveis linguísticos: o fonológico, o morfológico, o sintático e o semântico, contendo estruturas gramaticais próprias.

As línguas de sinais são formadas por um conjunto de sinais, os quais representam as palavras ou um item lexical de uma língua oral-auditiva. Os sinais são compostos por fonemas, que são as unidades básicas da comunicação baseada em sinais. Os sinais são formados por cinco unidades básicas que estão identificadas a seguir.

1 – Configuração das mãos: são as formas em que uma ou as duas mãos tomam durante a realização de um sinal. Exemplos de diferença de configuração da mão são a quantidade de dedos estendidos ou a contração da mão (aberta ou fechada).

2 – Ponto de articulação: é o local do corpo ou na frente do corpo onde o sinal é articulado. Os sinais podem ser articulados em um espaço neutro ou em alguma parte do corpo, por exemplo os sinais TRABALHAR, BRINCAR que são executados no espaço neutro enquanto os sinais APRENDER e PENSAR são feitos em um espaço do corpo, na testa.

3 – Movimento: um sinal pode ser composto por um movimento ou não e representa o deslocamento da mão. De acordo com Ferreira-Brito (1995), o parâmetro movimento é bastante complexo. Os tipos de movimento são: movimento interno da mão, movimento do pulso e o movimento direcional do espaço. Por exemplo, o sinal APRENDER, cujo movimento realiza-se com o abrir e fechar da mão, modificando sua configuração da mão.

4 – Orientação: De acordo com Quadros e Karnopp (2004), o parâmetro orientação é a direção para onde a palma da mão aponta quando produzimos o sinal e existem seis tipos de orientação da mão: para cima e para baixo, para dentro e para fora ou para os lados.

5 – **Expressões não-manuais**: um sinal pode ter a necessidade de características adicionais para expressar sentimentos de alegria, de tristeza, uma pergunta ou exclamação e é de grande importância porque sua utilização pode mudar completamente o significado de um sinal.

Conforme Buttussi et al. (2007), a partir da combinação desses cinco quiremas forma-se o sinal, que permitem comunicar diferentes ideias ou aspectos de uma sentença. Expressar-se em LIBRAS ou qualquer outra língua de sinais, portanto, é combinar esses elementos para formar os sinais e combinar os sinais para formar sentenças em um determinado contexto.

2.2.1 LIBRAS

A Língua Brasileira de Sinais (LIBRAS) é a língua de sinais utilizada no território brasileiro. Foi reconhecida e oficializada pela Lei n 10.436/02, de 24 de abril de 2002 e regulamentada pelo decreto 5626/2005.

O reconhecimento e oficialização da LIBRAS foi um grande avanço para a comunidade surda, pois além de outros avanços permitiu o acesso à educação e o direito de ter acesso à informação na sua língua nativa.

Como nas outras línguas de sinais, em LIBRAS os sinais são constituídos pelos cinco fonemas citados anteriormente, tem uma gramática própria e não deve ser comparada com a gramática da língua portuguesa. Com relação à ordem das palavras ou constituintes, por exemplo, existem diferenças entre a língua portuguesa e a LIBRAS. Enquanto que a língua portuguesa utiliza, na maioria dos casos, sentenças no formato sujeito-verbo-objeto (SVO), a LIBRAS utiliza, em geral, sentenças no formato tópico-comentário [Brito 1995].

2.3 Tradução automática

Traduzir é a transposição de um texto para uma língua-alvo de um conteúdo em uma língua-fonte tendo em mente sempre, durante o processo, que o objetivo final é gerar um texto na língua-alvo. (QUADROS; SOUZA, 2008:185).

A tradução automática é a conversão de conteúdos entre línguas naturais através de sistemas computacionais. Esse processo de conversão ou tradução, no entanto, possui uma série

de dificuldades e desafios intrínsecos. Por exemplo, quando um mensagem é transmitida de um interlocutor para outro, ela está, em geral, relacionada a um contexto e a um conjunto de conhecimentos de senso comum implícitos que precisam ser capturados e tratados pelo sistema de tradução para se gerar uma boa tradução. Além disso, as línguas naturais, em geral, possuem um conjunto de ambiguidades inerentes (ambiguidade léxica, sintática, semântica, contextual, etc.) que também precisam ser tratadas pelo sistema de tradução automática [Dorr et al. 1999].

De acordo com Dorr et al. (1999), esses desafios podem ser caracterizados em duas classes:

1. Desafios com relação aos aspectos linguísticos;
2. Desafios com relação aos aspectos operacionais.

Com relação aos aspectos linguísticos, os desafios podem ser classificados em três categorias: (1) compreensão de linguagem, (2) geração de linguagem e (3) mapeamento entre pares de linguagens. [Dorr et al. 1999].

2.3.1 **Suíte VLIBRAS**

O VLIBRAS é uma suíte de ferramentas computacionais de código aberto multiplataforma desenvolvida pelo LAVID/UFPB em parceria com a Rede Nacional de Pesquisa - RNP e o Ministério do Planejamento, Orçamento e Gestão, que permite a geração de conteúdos em LIBRAS. A suíte é composta pelas ferramentas VLibras-Desktop, VLibras-Plugin, VLibras-Video, VLibras-Núcleo, WikiLibras e um dicionário de sinais que alimenta todas as ferramentas. Nessa suíte os conteúdos são gerados a partir de textos, áudios, legendas e representados por um avatar 3D, chamado de Ícaro.

A ferramenta VLibras-Plugin, por exemplo, é um plugin de navegador que, quando habilitado, permite que o usuário traduza qualquer texto selecionado no navegador para LIBRAS de forma automática. Com isso, o usuário surdo pode navegar em qualquer página na Internet e acompanhar a sua tradução para LIBRAS. O VLibras-Desktop é uma ferramenta mais geral que o VLibras-Plugin e pode ser aplicada para traduzir textos selecionados em qualquer software ou aplicativo instalado no computador pessoal do usuário surdo para LIBRAS. Por fim, o VLibras-Video, como o próprio nome indica, tem como objetivo traduzir vídeos submetidos pelo usuário para LIBRAS, permitindo também que os usuários surdos tenham acesso a conteúdos multimídia na sua língua natural. Essas ferramentas possuem em comum um núcleo central de desenvolvimento, denominado VLibras-Núcleo, o qual tem como objetivo concentrar as principais funcionalidade dessas ferramentas. Nessa suíte, os conteúdos em

LIBRAS são gerados a partir da tradução automática de textos, legendas ou áudio em língua portuguesa e são representados por um agente animado virtual 3D (avatar-3D).

Para a geração desses conteúdos, um DaaS foi utilizado, juntamente com uma ferramenta Web de construção colaborativa de sinais em LIBRAS, denominada WikiLibras. A proposta do WikiLIBRAS é permitir que colaboradores possam participar do processo de desenvolvimento do Dicionário de LIBRAS através da adição de novos sinais ou da edição dos sinais existentes, tornando o seu desenvolvimento mais produtivo.

A suíte de ferramentas VLIBRAS vem sendo aprimorada há alguns anos pela equipe do LAVID através de apoios financeiros da RNP, CAPES, MPOG. Araujo (2012), Ferreira (2012) e Silva (2012) são alguns dos trabalhos que contribuem para o aperfeiçoamento do sistema.

Para realizar a tradução automática do Português para LIBRAS, a Suíte VLIBRAS utiliza alguns serviços que realizam consultas externas. Como exemplo de um serviço, existe o de tradução de um texto da língua portuguesa para a representação de um texto em LIBRAS, denominado glosa.

Segundo Grigório, et al. (2016), VLibras-Box é um componente da Suíte VLIBRAS e funciona como uma instância independente que incorpora todas as pilhas de serviços, devidamente orquestrados, necessárias para o processamento exibido para que uma requisição seja atendida. Cada VLibras-Box representa um servidor espelho utilizado para compor uma infra-estrutura de servidores redundantes e tolerantes a falhas.

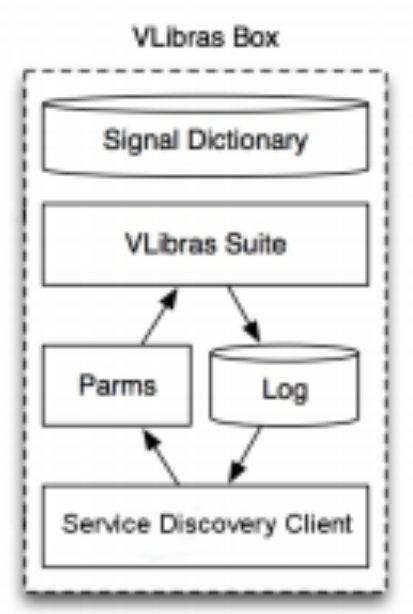

Figura 1: Componentes da VLibras-Box

Conforme a Fígura 1, pode-se observar a composição de uma VLibras-Box a qual é composta por uma cópia do dicionário de sinais em LIBRAS.

A VLibras-Suite reúne uma coleção de aplicações que se destinam a realizar cada um dos passos necessários para completar uma tradução Português-LIBRAS. O *Service Discovery Client* funciona como um cliente para algum serviço de descobrimento. Esse componente encapsula a comunicação entre a VLibras-Box e o *Service Discovery Server,* que é um servidor principal responsável pelo registro dos diferentes espelhos disponíveis na infraestrutura distribuída.

2.4 QD Framework

Segundo Pessoa et al. (2016), o QD Framework funciona como uma implementação funcional de uma arquitetura genérica para a construção de malhas de servidores espelho baseada em um catálogo dinâmico.

Através do QD Framework, o usuário final consegue localizar um melhor servidor dentro um malha de servidores que seja mais apropriado entre os ativos para atender a demanda de um determinado usuário. Para isso o QD leva em consideração fatores estáticos (e.g., geolocalização do servidor) e fatores dinâmicos (e.g., desempenho recente). Para isso, se faz o gerenciamento de uma malha de servidores espelho incluindo registro, catalogação, monitoramento, distribuição e acesso.

O QD Framework é dividido em quatro componentes principais, que são o QD Server, QD Node, QD Client e QD Monitor.

O QD Server realiza o gerenciamento da lista de servidores espelhos e a organiza dinamicamente para fornecer ao QD Client listas atualizadas e personalizadas, de acordo com critérios específicos. Já o QD Node faz a comunicação do servidor espelho com o QD Server para catalogação inicial e envio de informações sobre o desempenho. Em comunicação direta com o QD Server, o QD Client realiza a mediação entre o usuário e a origem do conteúdo desejado a partir do recebimento periódico de uma lista atualizada de servidores e aplicando heurísticas para escolher o servidor mais apropriado.

O QD Monitor é um componente de representação visual dos servidores espelho presentes em uma malha construída com o QD Framework. O QD Monitor foi projetado para ser facilmente incorporado a qualquer ambiente web ou portal e realiza comunicação direta com o QD Server para obter os dados sobre a malha de servidores.

2.5 Serviços Web

2.5.1 Interface de Programação Aplicada

Interface de Programação Aplicada (*Application Programming Interface* – API) é um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se nos detalhes da sua implementação, mas apenas utilizar seus serviços (Foldoc, 2015). Em essência, uma API define um modo como um desenvolvedor deve fazer requisições e requisitar algo para a sua aplicação.

As APIs oferecem um meio para desenvolver ou facilitar o desenvolvimento de uma aplicação. No caso dos sistemas operacionais, as funções permitem que os usuários acessem arquivos e criem janelas. No contexto web, por outro lado, elas geralmente são um conjunto definido de mensagens de requisição e resposta do tipo HTTP, HTTPS ou via WebSockets, geralmente no formato XML ou JSON.

Um exemplo de API *Webservice* que visa facilitar o desenvolvimento de novas aplicações é a API do Google Maps, a qual é um aplicativo desktop e web para explorar mapas, desenvolvido pela Google, que fornece uma API e permite que os desenvolvedores implantem-no dentro das suas aplicações, contando com uma série de funcionalidades para exploração dos mapas.

Outro exemplo é o Facebook, que fornece uma API com funções que permitem aos desenvolvedores a integração com suas aplicações, possibilitando aos usuários a integração de seus perfis autenticados e acesso à aplicação na rede social.

2.5.2 Node.js

O Node.js é um *framework* para o lado servidor que executa o V8 Javascript, a qual é compilada para código de máquina e executa em memória. Ele foi criado a partir de um interpretador ultra rápido escrito em C++ que a Google usa em seu navegador Chrome. Porém, é possível baixar e integrá-lo em qualquer aplicação que você desejar, não sendo restrito ao seu navegador. A utilização de Javascript no lado servidor da aplicação é um conceito relativamente novo. A ideia é a utilizar no lado servidor da aplicação a mesma linguagem utilizada no lado cliente.

O objetivo do Node é fornecer uma maneira fácil de desenvolver programas de redes escaláveis. Com relação a escalabilidade, em outras linguagens como Java e PHP cada nova conexão inicia um encadeamento que se aloca uma determinada quantidade de memória, o qual

é de aproximadamente 2MB. No caso de um sistema com 8GB de memória, estaríamos definindo o número máximo teórico de conexões simultâneas em cerca de 4000 usuários.

Diferentemente das outras linguagens, o Node não cria um novo encadeamento e, consequentemente, não há alocação de memória para novas conexões. Sua utilização é recomendada para situações em que um grande volume de tráfego é esperado e a lógica e o processamento necessários do lado do servidor não são necessariamente antes de responder o cliente.

Devido as suas características, ele é indicado para o desenvolvimento de aplicações que necessitam tratar milhares de requisições por segundo. Diferentemente do Apache ou Tomcat, o Node não é caracterizado como um servidor web. É possível, através desse *framework*, desenvolver um servidor com todas as funcionalidades necessárias, de forma personalizada.

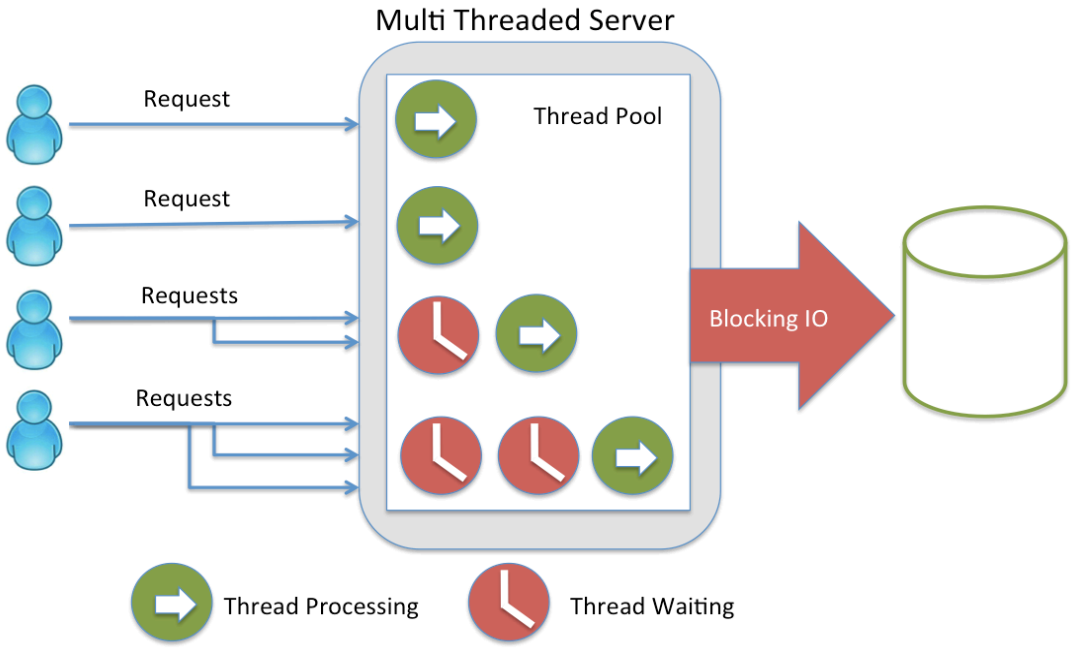

Figura 2: Modelo de conexão web multi-thread

Fonte: strongloop.com/node-js-is-faster-than-java/

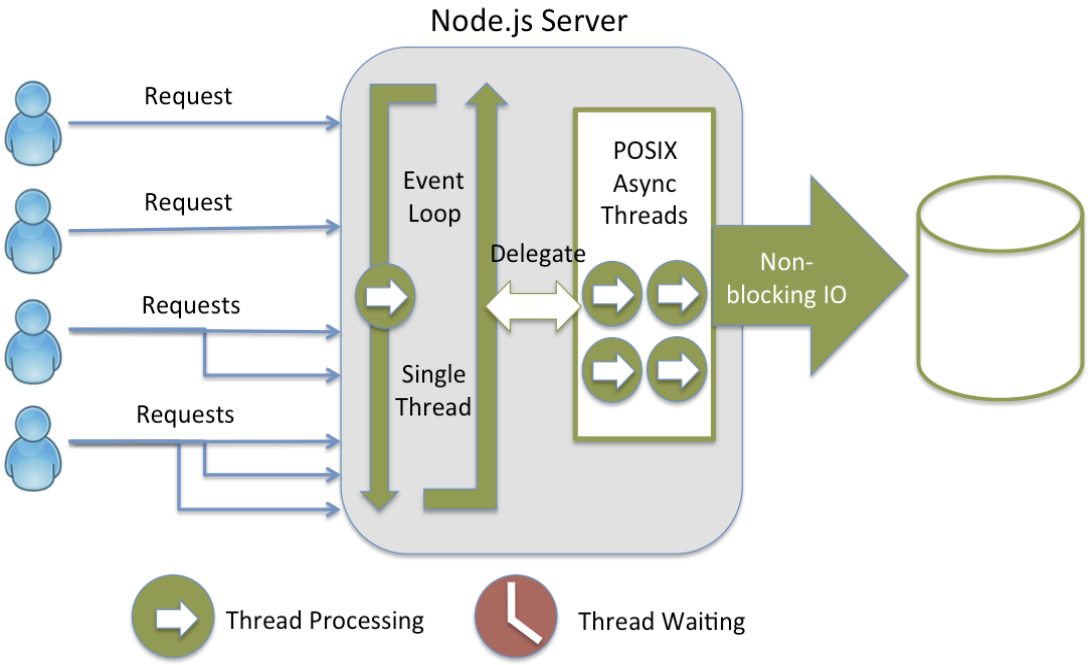

Figura 3: Modelo de conexões web single-thread

Fonte: strongloop.com/node-js-is-faster-than-java/

Na figura 2, pode-se observar a arquitetura de conexões de um servidor web, tal como Apache ou Tomcat. A cada requisição uma nova thread é criada e o servidor só consegue atender uma requisição por vez, gerando um enfileiramento e consequentemente aumentando o tempo de resposta. Já na Figura 2, tem-se a arquitetura de um servidor utilizando o Node.js, cujas requisições não são bloqueantes e o servidor continua executando novas requisições sem parar as demais e comunicando-se através de *callbacks*[2] para possíveis retorno de sucesso ou falha. Como exemplos de serviços da web que se há a utilização do Node, tem-se o Flickr e o Imgur. Esses serviços necessitam de um rápido processamento para oferecer respostas imediatas aos seus usuários e, deste modo, podem definir servidores dedicados para oferecer esse serviço.

Como exemplos de casos em que a utilização do Node não é recomendada, tem-se o caso de criação de páginas dinamicamente. Aplicações que contêm bancos de dados relacionais são outro exemplo de caso em que não se indica sua utilização. A explicação, neste caso, é que esses serviços são síncronos e com bloqueios e possuem objetivos diferentes dos do Node.js.

A LinkedIn, conhecida por ser uma das maiores redes sociais de negócios do mundo, migrou sua infra-estrutura móvel back-end antes em Ruby on Rails para Node.js por questões

[2] Os *callbacks* são funções que são passadas como parâmetro para ser executado quando outra função ou evento for concluído.

de problemas relacionados performance e escalabilidade. De acordo com Kiran Prasad, diretor de engenharia móvel do LinkedIn, a mudança fez com que em alguns cenários a resposta ficasse até 20 vezes mais rápida.

Outros exemplos de grandes empresas que utilizam o Node.js nas suas aplicações são a Netflix, New York Times, Paypal, Medium, Linkedln, Groupon, Flickr, eBay, Trello, Yahoo e Uber.

2.5.3 MongoDB

O NoSQL surgiu em 1998 como uma abreviação do termo "Not Only SQL" e surgiu como uma alternativa ou um complemento aos banco de dados relacionais. Eles são conhecidos como banco de dados não relacionais e visam resolver problemas atuais como escalabilidade, velocidade, adaptabilidade e consistência. As opções de NoSQL são aplicadas em armazenamento distribuído de dados e projetados para suportar requisitos de acesso a dados em grande escala.

A rede social Twitter chama atenção quando anunciou que estaria migrando do MySQL para o NoSQL Cassandra. Outras companhias conhecidas como Facebook, eBay e NASA também utilizam o NoSQL.

Um dos banco de dados NoSQL mais conhecidos é o MongoDB. O MongoDB é um banco de dados NoSQL que tem como proposta servir como *data storage* primário da aplicação, ou seja, ser usado no lugar de outros como MySQL, PostegresSQL e outros. Ele usa o modelo de dados flexível, de modo que o *schema* pode mudar facilmente conforme a aplicação evolui e, apesar de ser NoSQL, oferece funcionalidades comuns dos bancos de dados tradicionais como múltiplos índices, linguagem de consulta simples e operações de agregação e consistência.

No MongoDB os dados são armazenados em JSON, contendo atributos aninhados evitando normalizações desnecessárias. Ele foi construído para possuir escalabilidade, performance e alta disponibilidade, escalando desde implementações simples para grandes e complexas arquiteturas.

3 TRABALHOS RELACIONADOS

Neste capítulo serão apresentados os principais dicionários de LIBRAS digital. Além disso, também apresentamos algumas das principais ferramentas de tradução automática de Português para LIBRAS e uma análise sobre a disponibilidade do dicionário de sinais de cada ferramenta.

3.1 Dicionário Acesso Brasil

Em dezembro de 2002, o Instituto Nacional de Surdos – INES lançou um dicionário bilíngue – Português x Libras e Libras x Português utilizando as bases linguísticas e sinais filmados em movimento, sendo pioneiro no uso da tecnologia por apresentar de modo virtual um dicionário de LIBRAS. Além do pioneirismo, é importante ressaltar que ele é um dos dicionários de referência para LIBRAS no Brasil e é utilizado no apoio na educação de alunos surdos, como referência para pais e professores de surdos, especialistas e para a comunidade.

Para ter acesso aos sinais é possível requisitar um CD-Rom ao INES com os vídeos dos sinais ou através do site. No site os vídeos dos sinais são divididos por categorias e é possível buscar pelo nome de um sinal. O dicionário é antigo e não há informações sobre atualizações ou um controle de versão.

3.2 RYBENA

O Rybená é um recurso para os sites proverem acessibilidade em LIBRAS. Segundo o sítio oficial (www.rybena.org.br):

> "É um recurso de tecnologia assistiva que está preparado para funcionar de forma compatível com os principais navegadores, seja para computadores ou dispositivos móveis. Esta solução é capaz de traduzir textos do português para LIBRAS e de converter português escrito para voz falada no Brasil, oferecendo às pessoas com necessidades especiais a possibilidade do entendimento dos textos na internet.".

Uma limitação do sistema é que ele realiza a tradução das palavras da língua portuguesa diretamente para LIBRAS, sem análise de contexto.

No site do Rybená é possível encontrar uma lista com 41 (quarenta e um) clientes e parceiros que utilizam o recurso. Em alguns sites é possível ver o funcionamento do recurso.

Porém, em outros, embora constem como clientes do Rybená, não há a opção para ativar o recurso.

Não foi possível também obter a informação do valor que é cobrado para implementar o recurso no seu website. Também não foi possível encontrar informações acerca do dicionário de sinais utilizado pela ferramenta para realizar a tradução automática para LIBRAS.

3.3 POLI-LIBRAS

Segundo o sitio oficial do POLILIBRAS, esta ferramenta

> "consiste em um conjunto de projetos open-source desenvolvido por alunos do departamento de engenharia da computação da Escola Politécnica de São Paulo e tem por objetivo a criação de uma ferramenta de tradução automática que converta um texto em português para uma sequência de sinais em LIBRAS"

O POLI-LIBRAS propõe fazer uma tradução que leve em conta as diferenças estruturais sintáticas entre as línguas, de forma a gerar uma saída mais compreensível e não apenas uma tradução por palavra, conhecida como de "português sinalizado", como é o caso do Rybená. Assim como o VLIBRAS, o POLILIBRAS propõe a criação de uma WikiLibras como plataforma colaborativa para que a comunidade possa incluir novos sinais no seu dicionário. Os códigos da ferramenta estão disponíveis na ferramenta de controle de versões para código Github.

O dicionário da ferramenta POLI-LIBRAS é feito através da modelagem de sinais pela escrita de SignWriting. Os sinais são definidos em XMLs e o avatar em JavaScript processa e executa o sinal já definido.

Contudo, observa-se que há poucos sinais definidos junto ao código disponibilizado. Para utilizar o dicionário é necessário implementar os códigos do sistema dentro da própria aplicação.

3.4 TLIBRAS

Para realizar a tradução da língua portuguesa para LIBRAS, o projeto TLIBRAS (LIRA, 2003) consiste de duas etapas:

1. Tradução em tempo real do português para LIBRAS, a partir de uma entrada via texto ou voz;

2. Apresentação da animação resultante da tradução em salas de aula, TV Digital ou vídeos na internet (LIRA, 2009).

O processo de tradução do TLIBRAS utiliza-se da UNL (Universal Networking Language) (UCHIDA; ZHU, 2001) para criar uma língua intermediária e realizar a tradução. A UNL funciona como uma interlíngua em que todas as outras línguas podem ser traduzidas. Para isso foi necessário um mapeamento de diversos textos em LIBRAS para a UNL, com as informações morfológicas, sintáticas e semânticas. Esta ferramenta também utiliza um avatar 3D para visualização da tradução.

3.5 FALIBRAS

Coradine et al. (2004) propuseram o sistema Falibras com o intuito de auxiliar na comunicação entre ouvintes e surdos visando facilitar o convívio entre os mesmos. O sistema propôs captar uma entrada de voz e com a utilização dos recursos do IBM Viavoice converter a voz para texto. Após esse processamento, texto é analisado utilizando um interpretador léxico-morfologico-sintático a fim de analisar e definir o contexto de forma que se tenha uma tradução adequada em LIBRAS. Esse interpretador também realiza a escolha adequada das animações. Não foram encontradas informações sobre o dicionário de sinais do FALIBRAS para realizar a tradução automática para a LIBRAS.

3.6 PRODEAF

A Proativa (2013) define o PRODEAF como “um conjunto de soluções de software capazes de traduzir texto e voz de português para LIBRAS com o objetivo de permitir a comunicação entre surdos e ouvintes”. Ele utiliza-se do reconhecimento de voz para fazer a tradução e está disponível nas versões web e móvel.

A ferramenta de tradução automática dispõe o seu dicionário de sinais animados em um avatar 3D através do seu website. Para ter acesso ao dicionário é necessário autenticar-se com uma conta do *Facebook* e permitir acesso da aplicação aos dados do perfil.

Os sinais são disponibilizados apenas para visualização em ordem alfabética, por categorias ou uma lista de sinais colaborativos. Estes sinais colaborativos são criados pelos usuários e são agrupados por estado, região ou como sinais internacionais. O site da ferramenta esclarece que no dicionário pode existir sinais incorretos, pois foram criados pelos usuários e não foram validados.

Na página oficial da ferramenta não há nenhuma indicação na página da ferramenta se é possível ter acesso aos arquivos dos sinais em LIBRAS utilizado pela ferramenta para o seu processo de tradução automática para LIBRAS nem mesmo como isso pode ser feito.

3.7 HANDTALK

A Handtalk realiza a tradução digital e automática para a Língua de Sinais, utilizada pela comunidade surda. A solução oferece ferramentas complementares ao trabalho do intérprete para auxiliar a comunicação entre surdos e ouvintes através de um avatar 3D. O aplicativo está disponível para as plataformas web e móvel, além de fazer tradução de vídeos.

Não foram encontradas informações sobre o funcionamento do dicionário de sinais da ferramenta. A Fígura 4 demonstra a ferramenta do Handtalk para tornar páginas da web acessíveis, em que a ferramenta é incorporada a página do usuário do sistema.

Figura 4: Ferramenta do Handtalk para tornar sites acessíveis
Fonte: http://www.handtalk.me/app

3.8 Quadro comparativo

Para melhor análise das ferramentas descritas nas seções anteriores a Tabela 1 apresenta as ferramentas de tradução automática levantadas e informações sobre seus respectivos dicionário de sinais. A maioria das ferramentas não dispõe informações sobre seu dicionário de sinais.

Ferramenta	**Dicionário de sinais**
Dicionário Acesso Brasil	Dicionário aberto, é possível adquirir o dicionário em formato de vídeo em um CD-Rom.
RYBENA	Não foram encontradas informações sobre.
TLIBRAS	Não foram encontradas informações sobre
FALIBRAS	Não foram encontradas informações sobre.
PRODEAF	Possível visualizar o dicionário por uma página na web, porém não há informações de como obter uma cópia do dicionário completo.
HANDTALK	Não foram encontradas informações sobre.

Tabela 1: Relação de ferramentas e informações sobre os seus respectivos dicionário de sinais

4 SOLUÇÃO PROPOSTA

Conforme mencionado no capítulo 3, existem um conjunto de ferramentas de tradução automática que buscam ajudar a resolver o problema de comunicação dos surdos. No entanto, essas ferramentas possuem dicionário de sinais fechados e não oferecem possibilidade de integração com aplicações de terceiros, não provendo soluções para o problema descrito neste trabalho.

Para buscar solucionar o problema e cumprir com os objetivos deste trabalho, propomos o serviço de uma API Webservice pública com o dicionário da Suíte VLibras. Para melhor explicar o serviço proposto neste capítulo, serão apresentados os requisitos funcionais e não funcionais, a arquitetura do serviço, a proposta de funcionamento da API do serviço e as tecnologias utilizas para o desenvolvimento.

4.1 Requisitos

Para definir as funcionalidades do serviço, os possíveis cenários de uso foram levantados e definidos juntamente com uma equipe de desenvolvedores do projeto Suíte VLIBRAS. O objetivo do levantamento foi identificar os principais requisitos funcionais e não-funcionais associados ao gerenciamento de versões do dicionário do VLIBRAS, a consulta, publicação de sinais, replicação, provimento de funcionalidades para terceiros, dentre outros.
Sommerville (2008) classifica os requisitos de sistema de software como funcionais, não funcionais e como requisitos de domínio:

- Requisitos funcionais definem as funcionalidades do sistema como deve reagir em condições específicas e como se comportar em determinadas situações. Podem ainda declarar o que o sistema não deve fazer;
- Requisitos não funcionais são restrições sobre serviços ou funções oferecidas pelo sistema. Dentre elas destacam-se restrições de tempo, sobre o processo de desenvolvimento e de padrões. A descrição das restrições complementa a definição de requisitos (PAULA FILHO, 2000);
- Requisitos de domínio são restrições originárias do domínio da aplicação do sistema e refletem características do mesmo. Podem ser requisitos funcionais ou não funcionais.

4.1.1 Requisitos Funcionais

A partir do levantamento com uma equipe de desenvolvedores do VLIBRAS, foram elaborados os seguintes requisitos funcionais para o DaaS (ver Tabelas de 1 a 5).

Tabela 2: Requisito funcional “Listar sinais” do DaaS

Identificador	[RF01]
Nome	Listar sinais
Prioridade	Essencial
Descrição	Permite que o usuário liste todos os sinais cadastrados no serviço. Além de listar todos os sinais cadastrados o usuário também pode pesquisar opcionalmente os sinais que iniciam com uma determinada letra que deve ser enviada ou buscar por uma versão específica do dicionário.
Pré-Condições	Realizar uma consulta no serviço.
Pós-Condições	O sistema deve retornar a lista com todos os sinais ou filtrados de acordo com o especificado na consulta e a qual versão do dicionário ele está associado.
Fluxo Principal [FP]	1. Usuário faz requisição ao serviço. 2. Serviço processa a requisição e valida o seu tipo. 3. Serviço consulta seu banco de sinais cadastrados de acordo com tipo da requisição. 4. Serviço retorna para o usuário a lista de sinais.
Fluxo Secundário [FS]	1. Banco de sinais indisponível 2. Sistema retorna erro informando que não foi possível conectar com o banco de sinais.

Tabela 3: Requisito funcional "Consultar sinal" do DaaS

Identificador	[RF02]
Nome	Consultar sinal
Prioridade	Essencial
Descrição	Permite que o usuário consulte a existência de um sinal especifico no sistema.

Pré-Condições	Realizar uma consulta ao serviço.
Pós-Condições	O sistema deve retornar informando a existência ou não do sinal passado ao usuário.
Fluxo Principal	1. Usuário faz requisição ao serviço 2. Serviço processa requisição e consulta no banco de sinais o sinal enviado. 3. Serviço retorna informando sobre a existência do sinal.
Fluxo Secundário [FS]	1. Banco de sinais indisponível 2. Sistema retorna erro informando que não foi possível conectar com o banco de sinais.

Tabela 4: Requisito funcional "Consultar versão do Dicionário" do DaaS

Identificador	[RF03]
Nome	Consultar versão do dicionário
Prioridade	Essencial
Descrição	Permite que o usuário consulte a versão atual do dicionário da máquina.
Pré-Condições	Realizar uma consulta ao serviço.
Pós-Condições	O sistema deve retornar a versão atual do dicionário do sistema.
Fluxo Principal [FP]	1. Usuário faz requisição ao serviço. 2. Serviço consulta a sua versão do dicionário. 3. Serviço retorna a sua versão do dicionário.

Tabela 5: Requisito funcional “Baixar sinais” do DaaS

Identificador	[RF04]
Nome	Baixar sinais
Prioridade	Essencial
Descrição	Permite que o usuário obtenha sinais cadastrados no sistema. O usuário poderá obter um sinal específico ou sinais de uma versão específica além de todos os sinais cadastrados no sistema. O usuário também deve poder enviar a sua versão do dicionário e receber um arquivo compactado com todos os sinais novos se houverem.
Pré-Condições	Realizar uma consulta ao serviço.

Pós-Condições	Serviço retorna sinais compactados.
Fluxo principal [FP]	1. Usuário faz requisição serviço. 2. Serviço processa requisição. 3. Serviço retorna os sinais compactados.
Fluxo Secundário [FS]	1. Usuário faz requisição ao serviço. 2. Serviço processa a requisição. 3. Parâmetro inválido é detectado. 4. Serviço retorna que houve erro na requisição informando que um dos campos é inválido.

Tabela 6: Requisito funcional "Listar Requisições a sinais" do DaaS

Identificador	[RF04]
Nome	Listar requisições
Prioridade	Essencial
Descrição	Permite que o usuário liste todas as requisições a um ou vários sinais. O usuário poderá obter as requisições bem sucedidas e falhas, além de que poderá ordenar por nome ascendente ou decrescente.
Pré-Condições	Realizar uma consulta ao serviço.
Pós-Condições	Serviço retorna nome do sinal, se o sinal existe e a quantidade de requisições efetuadas ou de vários de uma vez.
Fluxo Principal [FP]	1. Usuário faz requisição ao serviço. 2. Serviço processa a requisição. 3. Serviço retorna lista de sinais.
Fluxo Secundário [FS]	1. Usuário faz requisição ao serviço. 2. Serviço processa a requisição. 3. Sinal não é encontrado na lista de requisições efetuadas. 4. Serviço retorna ao usuário informando que nenhuma requisição referente ao sinal foi encontrada.

Tabela 7: Requisito funcional "Cadastrar sinal" do DaaS

Identificador	[RF06]
Nome	Cadastrar sinal

Prioridade	Essencial
Descrição	Usuários poderão cadastrar um sinal no sistema.
Fluxo Principal [FP]	1. Usuário faz requisição ao serviço. 2. Serviço processa a requisição. 3. Serviço envia consulta de cadastro do sinal no banco de sinais. 4. Serviço realizada procedimento de chamada ao repositório central. 5. Serviço retorna ao usuário que o sinal foi cadastrado ou atualizado.
Fluxo Secundário [FS01]	1. Usuário faz requisição ao serviço. 2. Serviço processa a requisição. 2. Serviço detecta que nome do sinal não foi enviado 3. Serviço retorna ao usuário que o nome do sinal deve ser enviado.
Fluxo Secundário [FS02]	1. Usuário faz requisição ao serviço. 2. Serviço processa a requisição. 2. Serviço detecta que arquivo blender não foi enviado. 3. Serviço retorna ao usuário que arquivo blender deve ser enviado.

4.1.2 Requisitos Não-Funcionais

Identificador	[RNF01]
Nome	Tipo de retorno esperado
Requisitos funcionais relacionados	Consultar versão do dicionário Consultar sinal Listar sinais
Prioridade	Essencial
Descrição	Para consulta a versão do dicionário, listar os sinais ou a existência de um sinal o usuário deverá informar o tipo de retorno esperado em JSON ou XML.

Tabela 8: Requisito não-funcional "Restrição para consultas" do DaaS

Identificador	[RNF02]
Nome	Arquivo e nome do sinal
Requisitos funcionais relacionados	Cadastrar sinal
Prioridade	Essencial
Descrição	Para cadastrar um sinal no serviço o usuário deve enviar um arquivo do tipo blender e o nome do sinal.

Tabela 9: Requisito não-funcional "Restrição para cadastro de sinais" do DaaS

Identificador	[RNF03]
Nome	Tipo de formato
Requisitos funcionais relacionados	Baixar sinais
Prioridade	Essencial
Descrição	Para baixar sinais pelo serviço, o usuário deverá especificar em que tipo de formato deseja receber os sinais, seja em vídeo, blender ou animação para Unity.

Tabela 10: Requisito não-funcional "Restrição para baixar sinais"

4.2 Arquitetura do Serviço

A arquitetura foi definida pensando na escalabilidade, em reduzir possíveis gargalos de sobrecarga. Conforme a Figura 5, o DaaS será executado sobre as instâncias da VLibras-Box. Cada VLibras-Box representa um servidor espelho utilizado para compor uma infraestrutura de servidores redundante e tolerante a falhas. O VLibras-Box funciona como uma instância independente que incorpora toda as pilhas de serviços, devidamente orquestrados para solucionar a demanda do processamento exigido para que uma tradução da suíte VLIBRAS seja atendida. Além disso, cada VLibras-Box possui uma cópia do dicionário de sinais e mantém

sincronia com o repositório central de sinais.

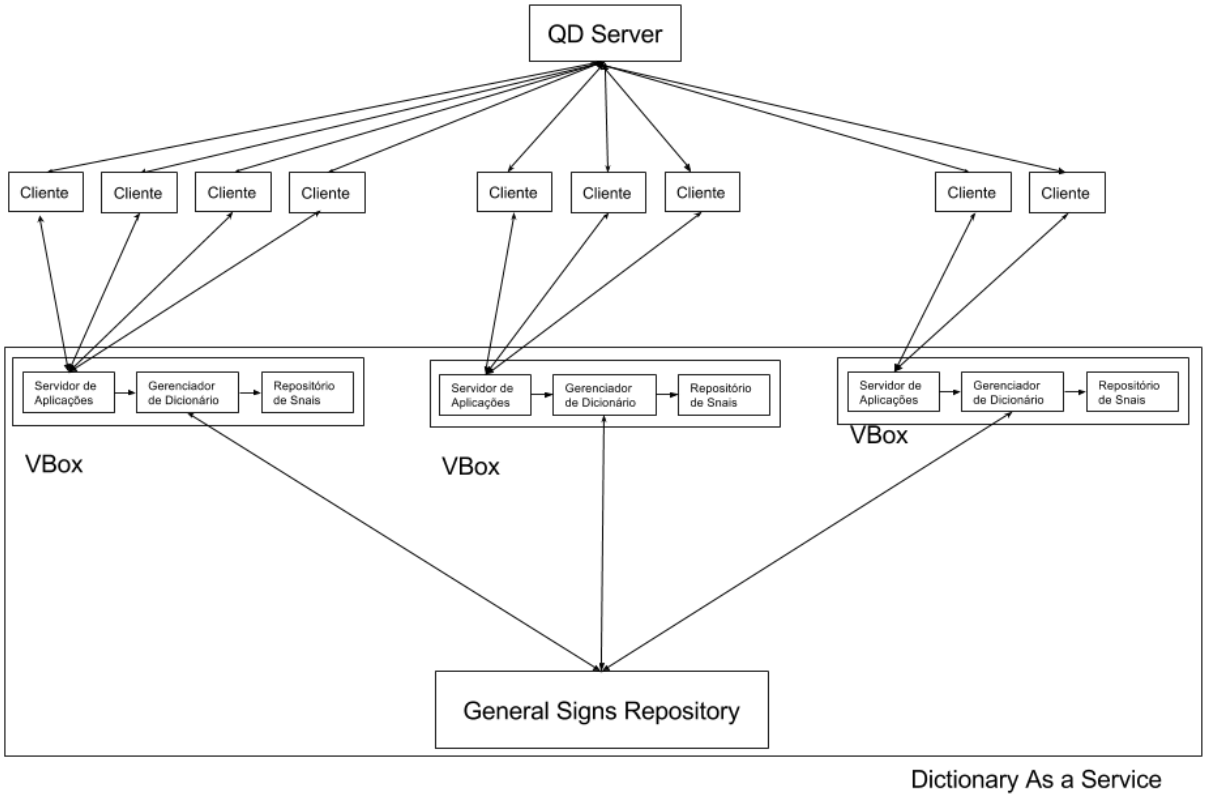

Figura 5: Arquitetura do DaaS

Conforme observado na Figura 5, inicialmente, todos os clientes do serviço devem fazer requisições ao QD Framework para obter o endereço de uma VLibras-Box.

O QD Client é o mediador entre usuário e origem do conteúdo desejado, sendo responsável pela seleção de qual espelho deve ser utilizado pelo usuário. Ele deve ficar embarcado na aplicação cliente do serviço e mantém uma lista de VLibras-Box representado pelo nome VBox na Fígura 5 e suas características. Para escolher a VLibras-Box que melhor pode atender as suas requisições, pode-se solicitar diretamente ao QD Server ou caso o QD Server esteja indisponível, pode-se utilizar heurísticas definidas no próprio QD Client o qual mantém uma cópia da lista de VLibras-Box ativas.

Os usuários realizam requisições ao serviço através de um servidor de aplicações que contém uma API *Webservice* embarcada. As requisições devem serem realizadas através dos *endpoints* da API, os quais definem as funcionalidades do serviço. Essas requisições podem ser de busca, recuperação, remoção ou inserção.

Em caso de requisições de busca, o serviço deve consultar apenas a sua cópia local do dicionário para atender ao usuário. Nas requisições de remoção ou inserção de sinais, o DaaS

realizada uma chamada ao repositório central para atualizá-lo, cabendo ao repositório refletir esta atualizações em todas as VLibras-Box a partir do envio de notificações às instâncias acerca da atualização de suas versões.

Todas as requisições são registradas no banco de dados com a finalidade de prover um meio de administrar o serviço, analisando os requisições mais realizadas, as requisições não respondidas, os sinais que foram mais baixados, entre outros.

Para controle da sua cópia do dicionário de sinais, o DaaS realiza verificações acerca de possíveis atualizações no repositório central. Ao detectar uma atualização, os arquivos novos contendo os novos sinais cadastrados ou atualizados no serviço são baixados, separados em pastas e cadastrados no banco de dados. Após cadastrar no banco de dados, são gerados arquivos compactados com todos os sinais correspondentes a nova versão nos formatos disponibilizados. Além de criar um arquivo compactado com os sinais da nova versão, o DaaS gera novos arquivos compactados com todos os sinais do dicionário.

Conforme representado na Figura 5, cada servidor espelho que faz parte da infraestrutura do DaaS mantém uma cópia local do repositório de sinais, o qual mantém uma comunicação com o repositório central, chamado *General Signs Repository*.

4.3 Dictionary As a Service

O Dictionary As a Service, serviço proposto neste trabalho, é elaborado baseado no modelo de disponibilização de uma API *webservice*. Com a utilização do DaaS os desenvolvedores podem criar suas aplicações utilizando o dicionário do serviço, que está na nuvem. O serviço proposto utiliza o modelo de arquitetura REST, que leva o nome de RESTful ao ser aplicado em APIs webservice fazendo uso do padrão URI. O padrão REST faz uso dos protocolos HTTP/HTTPS para comunicação entre os lados cliente e servidor. Os recursos individuais enviados são identificados na própria URI da requisição, possibilitando uma integração com diversas plataformas.

Algumas das vantagens de utilização do DaaS são listadas abaixo:

- Possibilidade de utilizar, na sua aplicação, um dicionário com mais de 8 mil sinais, modelados por uma equipe de animadores em um avatar 3D e avaliados por especialistas da LIBRAS;
- Trata-se de uma API pública, aberta e qualquer pessoa pode utilizar sem qualquer ônus;
- Os sinais são disponibilizados pelo serviço em vários formatos. Ao fazer a requisição ao serviço, o tipo de retorno do sinal deve ser informado (vídeo, animações, metadados, entre outros), permitindo que o usuário defina qual é o formato mais apropriado pra sua aplicação. Com isso, é possível portar o dicionário para diferentes tecnologias como, por exemplo, computadores pessoais (*desktop*), plataformas *web*, dispositivos móveis, aplicações para TV Digital e Cinema Digital;
- Possibilidade dos desenvolvedores realizarem um controle de versão de sinais individualmente ou a partir de um pacote de sinais para manter a aplicação em sincronia com o serviço;
- Os sinais ofertados tem origem da Suíte VLibras, a qual caracteriza-se pode ser uma suíte de ferramentas de tradução automática em produção. Dessa forma, há uma expectativa de que esse dicionário seja atualizado frequentemente, a partir de atualizações produzidas pela equipe responsável pela suíte.
- O dicionário também poderá ser atualizado pela comunidade surda, a partir do WikiLibras. Como citado anteriormente, o WikiLibras é uma ferramenta colaborativa em que a comunidade pode sugerir correções em sinais existentes ou propor a adição de novos sinais no sistema. Desse modo, há uma maior participação da comunidade na criação e manutenção do dicionário de sinais, pois é possível, através do WikiLibras, uma maior contribuição da sociedade na sugestão de correções e criação de novos sinais;
- Ele foi elaborado para suportar uma grande demanda de uso, utilizando uma arquitetura flexível e de fácil integração, possibilitando aos usuários do sistema a

criação de servidores espelhos e inclusão na lista de servidores disponíveis para dispor a toda a comunidade.

4.4 Tecnologias Utilizadas

Conforme mencionado na Seção 1.1.1, a proposta deste trabalho é a criação de um serviço para gestão de um Dicionário de Libras que atenda a demanda da suíte de aplicações de ferramentas automáticas VLIBRAS, além de aplicações de usuários terceiros. Nesse contexto, as tecnologias Node.js e MongoDB foram escolhidas para funcionar como servidor de aplicações e banco de dados para o serviço proposto.

A justificativa para a escolha do Node.js foi o fato dele permitir a realização de várias requisições simultâneas, baixo uso de CPU e memória, além do fato de ser uma das tecnologias mais utilizadas no mercado para a construção de APIs Restful.

O MongoDB foi escolhido por se tratar de um banco de dados escalável, de rápido acesso, consistente, além de possuir fácil integração com o Node.js. Para fazer a integração do Node.js utilizamos o Moongose. No MongoDB são armazenados todos os nomes dos sinais e a sua versão correspondente.

5 RESULTADOS E DISCUSSÕES

Neste capítulo, inicialmente será abordada a integração do serviço com o VLibras, utilizada para validar algumas funcionalidades do serviço proposto. Em seguida, serão apresentados e discutidos alguns testes realizados para analisar o desempenho do sistema em alguns cenários de uso.

5.1 Integração com o VLibras

Como informado nos capítulos anteriores, a proposta deste trabalho é que o serviço de dicionário de LIBRAS aberto proposto seja utilizado na gestão e provimento distribuído de um dicionário de LIBRAS para a Suíte VLIBRAS, além de oferecer um serviço aberto para facilitar e, consequentemente, estimular a criação de aplicações com tradução automática para LIBRAS. Na Figura 5, é ilustrado o processo de integração do DaaS com as ferramentas da Suíte VLibras.

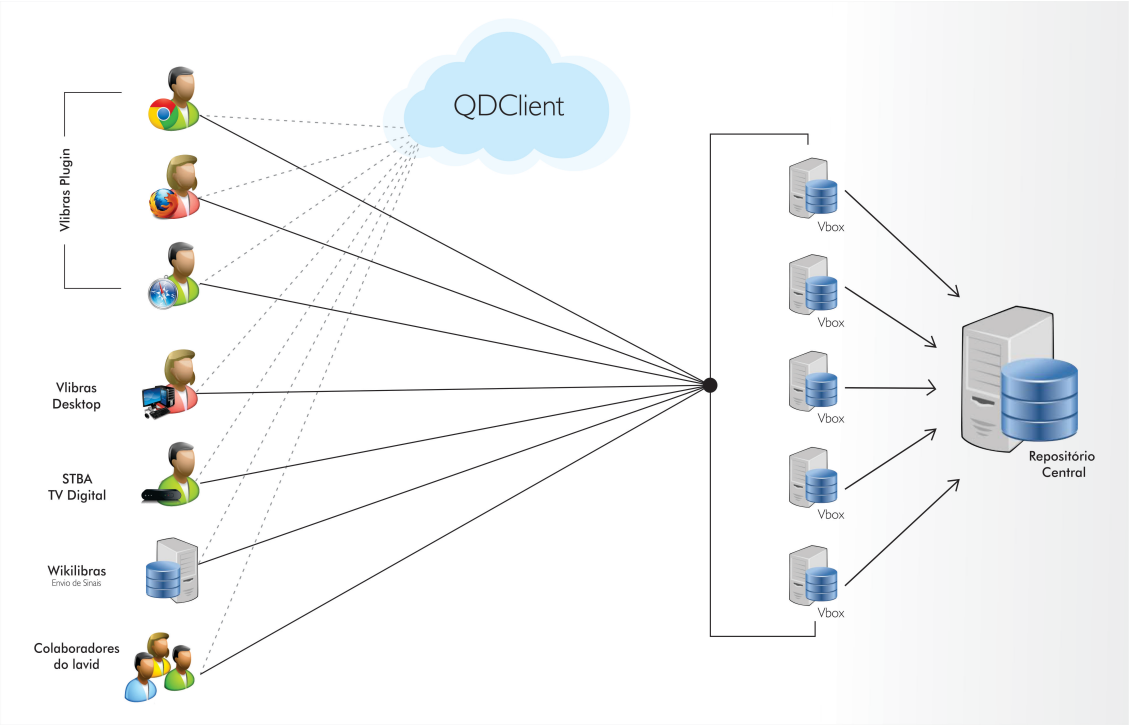

Figura 6: Representação da Integração do DaaS com a Suíte VLIBRAS

De acordo com a Figura 6, é possível observar algumas funcionalidades do DaaS integradas ao VLIBRAS. A partir dessa integração, torna-se possível validar algumas das funcionalidades propostas do serviço. Por exemplo, o VLibras-Plugin representado na imagem por usuários utilizando navegadores comuns no mercado, como o *Google Chrome, Mozilla*

Firefox e *Safari,* conecta-se ao DaaS, em tempo real para baixar os sinais requisitados pelo o usuário durante a tradução do conteúdo de um site. Nesse caso, é possível explorar a utilização da funcionalidade de *download* de sinais específicos sob demanda em que se envia ao DaaS o nome do sinal e em que tipo o sinal deve ser retornado.

Diferente do caso anterior em que os sinais são baixados por demanda, na versão VLibras-Desktop, os usuários têm a sua cópia local do dicionário e essa cópia possui uma versão associada. A partir do DaaS, é possível fazer um controle de versão, o que possibilita a atualização do dicionário para a versão mais recente. Mais especificamente, esse controle de versão faz uma comparação entre a versão do dicionário local do usuário e a última versão do dicionário disponível no serviço, permitindo que a aplicação notifique o usuário que existem atualizações disponíveis e o usuário possa atualizar a sua cópia local do dicionário. Para isso, o VLibras-Desktop envia ao serviço a sua versão atual do dicionário e o serviço verifica se há novas versões do dicionário. Se houver, ele retorna um arquivo compactado contendo a diferença entre as duas versões do dicionário.

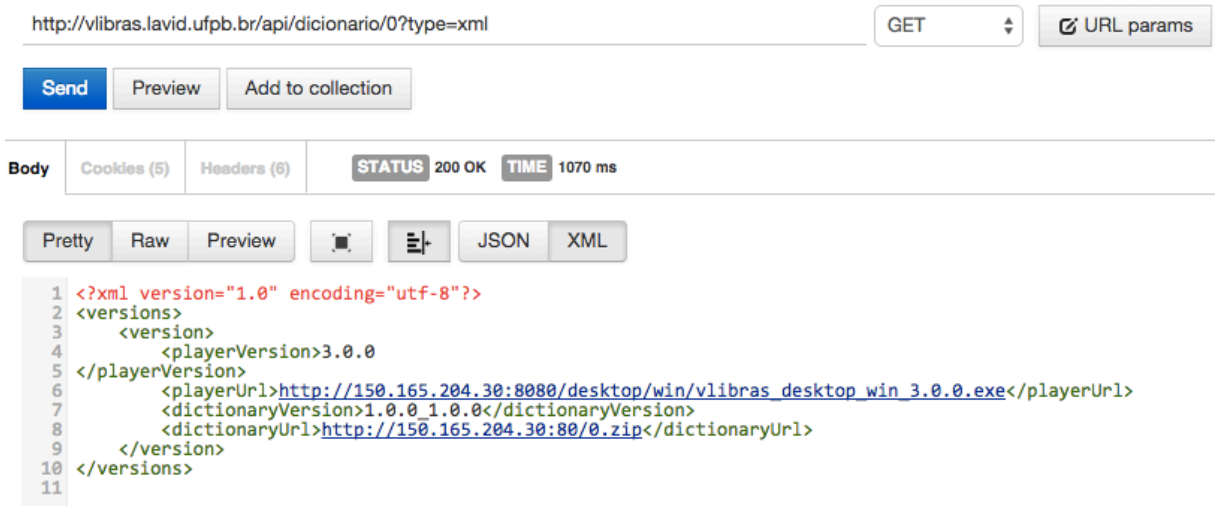

Figura 7: Exemplo de requisição com retorno no formato JSON utilizando o POSTMAN

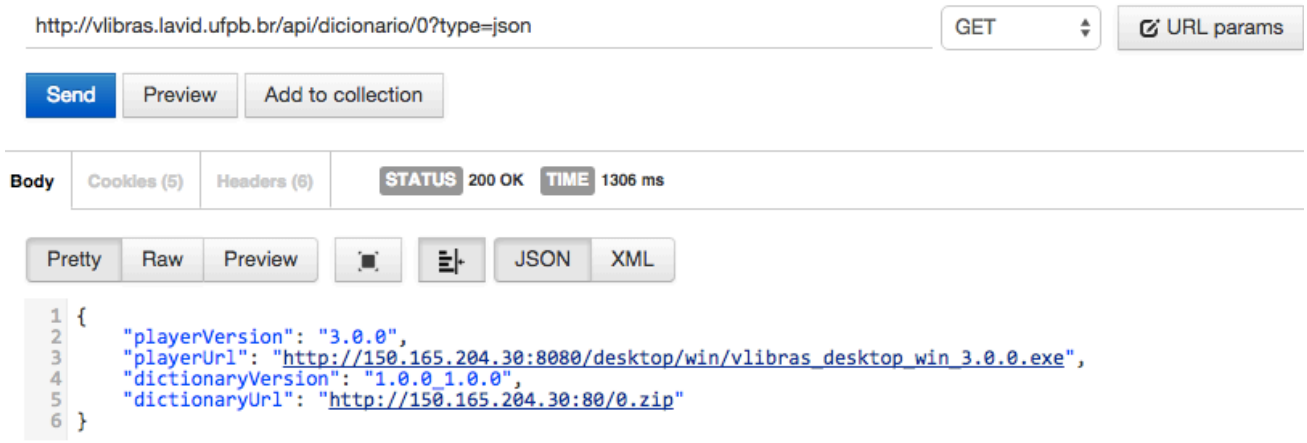

Figura 8: Exemplo de requisição com retorno no formato XML utilizando o POSTMAN

Nas Figuras 7 e 8, é possível visualizar o retorno nos formatos JSON e XML da consulta pela versão do dicionário utilizada pelo VLibras-Desktop, que deve realizar requisições do tipo *GET* enviando como parâmetros a sua versão atual do dicionário e o tipo de retorno esperado. Caso a versão do dicionário local seja inferior a última versão disponível no serviço, o DaaS cria então um arquivo compactado com a diferença entre as duas versões, e o retorna no campo *dictionaryUrl*. O VLibras-Desktop então verifica o retorno do DaaS e informa ao usuário que há novas atualizações disponíveis. Caso o usuário opte por atualizar a sua cópia do dicionário, o arquivo enviado no campo *dictionaryUrl* então é baixado e descompactado na pasta de sinais. Em seguida, a versão da cópia local do dicionário é atualizada para a versão informada no campo *dictionaryVersion*.

Para representar os usuários que podem atualizar o repositório de sinais do sistema, utilizamos na Figura 6 os papéis dos Colaboradores do Lavid ou o WikiLibras. Os colaboradores do LAVID podem adicionar diretamente, usando a função disponível para isso, de forma manual. O WikiLibras é uma ferramenta colaborativa que possibilita à comunidade, representada pelos usuários em uma interface amigável, ajudar a melhorar o dicionário de sinais da Suíte VLIBRAS. Dessa forma, as alterações requisitadas pelos usuários serão relacionadas a novas versões do sistema. Para atualizar o repositório, os colaboradores do Lavid ou o WikiLibras enviam ao DaaS um arquivo de animação no formato *Blender*. O arquivo é então enviado ao repositório central, convertido para todos os formatos utilizados pelos serviços (animações, vídeos, metadados, etc.). Em seguida, a versão do Dicionário é atualizada e essas mudanças se refletem nos demais servidores espelho conectados neste repositório.

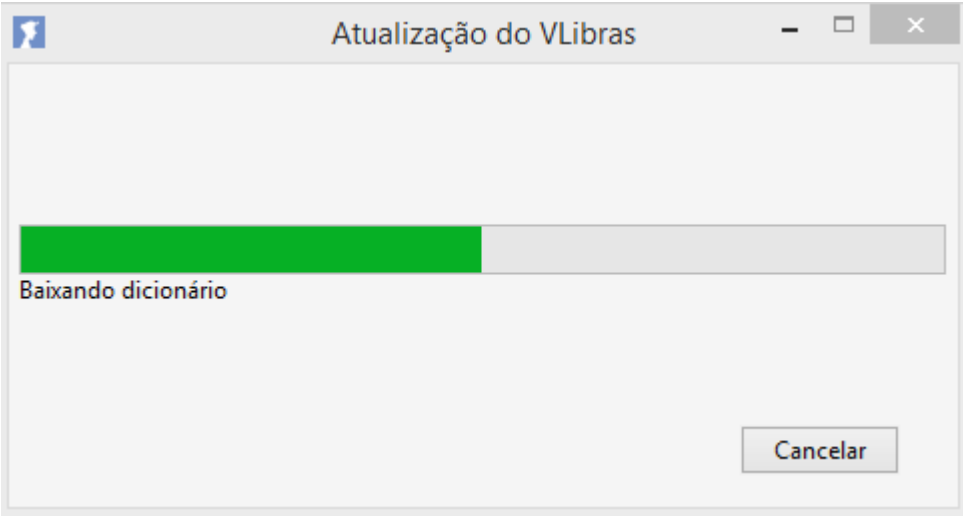

Figura 9:Tela de atualização do dicionário do VLibras-Desktop Windows

Após ser instalado, o VLibras-Desktop realiza o download de todos os sinais disponíveis no serviço. Na Figura 9, é possível observar a primeira tela de atualização do sistema realizando o download de todos os sinais disponíveis.

Após essa etapa, o usuário pode verificar, há qualquer momento, se existem novas atualizações disponíveis, conforme apresentado na Figura 10. Durante essa verificação, uma consulta ao serviço é realizada com objetivo de comparar as duas versões do dicionário (a versão local do dicionário do usuário e a última versão disponível no serviço). Caso o dicionário esteja atualizado, a mensagem ilustrada na Figura 10 será apresentada ao usuário.

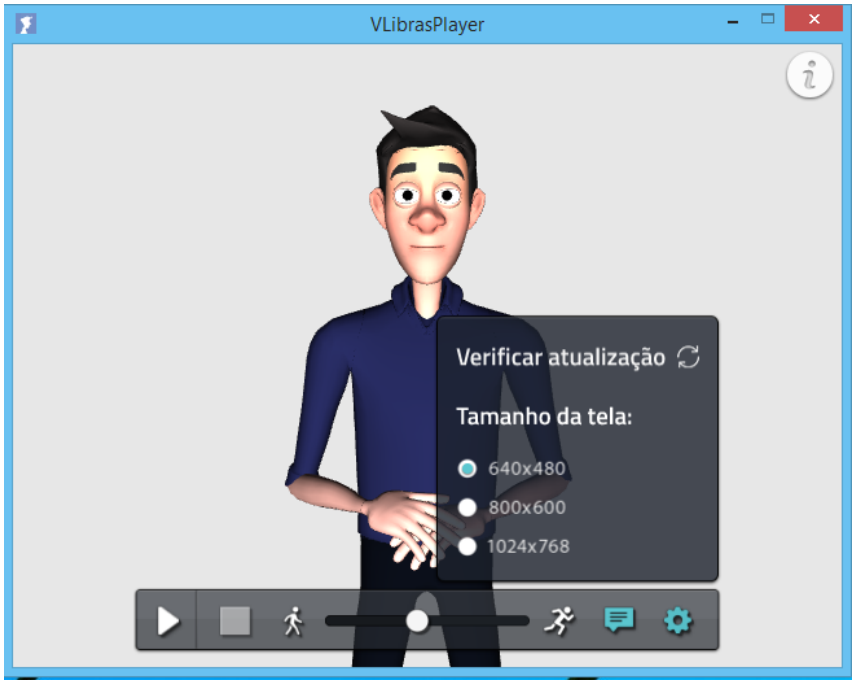

Figura 10: Tela principal do VLibras-Desktop Windows com opções de configuração

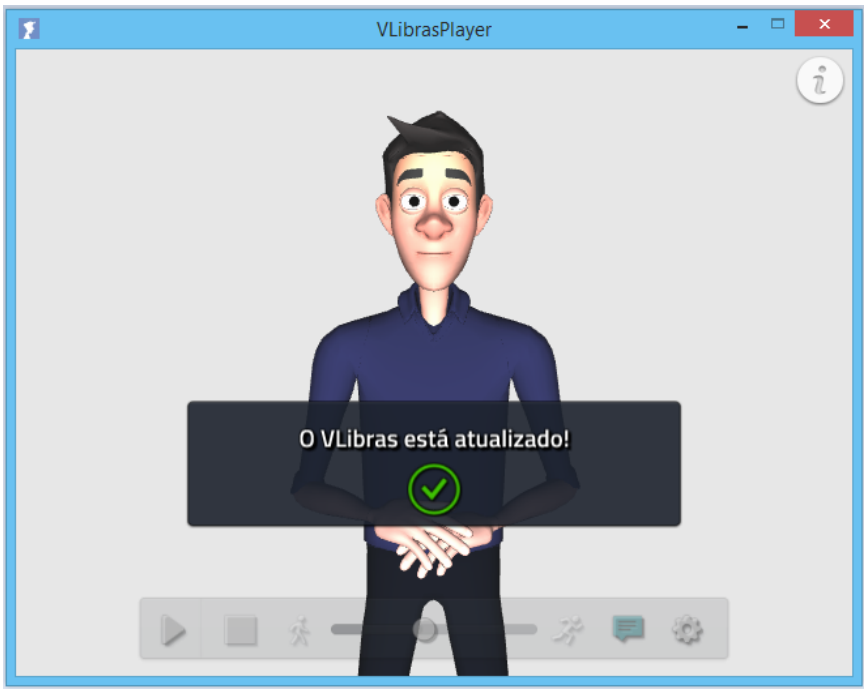

Figura 11: Tela principal do VLibras-Desktop informando sobre atualização

5.2 Testes Computacionais

Com o objetivo de identificar e avaliar o desempenho do DaaS, foram realizados testes computacionais utilizando a ferramenta JMeter. O JMeter é uma ferramenta criada pela fundação Apache para execução de testes de carga e *stress*. Através dos testes computacionais, pretende-se obter os possíveis gargalos e avaliar a capacidade do sistema. A proposta é que o DaaS esteja em uma arquitetura flexível com diversos servidores. Porém, neste primeiro momento os testes foram realizados utilizando apenas um servidor específico para avaliar a sua capacidade.

Os testes foram realizados utilizando a máquina que atualmente está em produção para suprir a demanda de sinais das ferramentas da Suíte VLIBRAS, a qual possui um processador Intel(R) Xeon(R) CPU E5620 2.40GHz com 4GB de memória RAM. Para isso, os testes foram estruturados de acordo com a funcionalidade a ser validada. Inicialmente, foi verificada a funcionalidade de realizar o download dos sinais sob demanda. Em seguida, foram avaliadas as funcionalidades de realizar o download de um conjunto de sinais por versão e de consulta da versão do dicionário. É importante ressaltar que os testes foram realizados fora da rede em que

a máquina de produção do VLIBRAS está hospedada para que o resultado se aproximasse de uma situação de uso real.

5.2.1 Testes com sinais específicos sob demanda

O teste da funcionalidade de realizar o download dos sinais sob demanda, que é utilizado por aplicações que necessitam dos sinais em tempo de execução como as extensões para navegadores, foi realizado a partir do envio de um conjunto de requisições simultâneas para baixar sinais específicos. O tamanho dos arquivos dos sinais pouco oscilam, então para este teste foi escolhido o sinal com um tamanho mediano dentre os vistos, o sinal escolhido foi o sinal que representa a palavra ABACATE.

Na Tabela 11 e na Figura 12, são apresentadas as taxa de erro, taxa de transferência e tempo médio de resposta obtidos para 1, 100, 400, 700, 1.000, 2.000 e 4.000 requisições realizadas simultaneamente no serviço para este tipo de funcionalidade.

Tabela 11: Informações do teste de carga e desempenho da funcionalidade de sinais sob demanda

Número de Requisições	**Taxa de falha**	**Taxa de transferência**	**Tempo médio de resposta**
1	0%	1100,98kb/sec	699ms
100	0%	1005,98kb/sec	2153ms
400	0%	680,15kb/sec	6263ms
700	0%	650,21kb/sec	10043ms
1000	0,27%	615,48kb/sec	15248ms
2000	24,31%	598,79kb/sec	12371ms
4000	33,48%	530,21kb/sec	111123ms

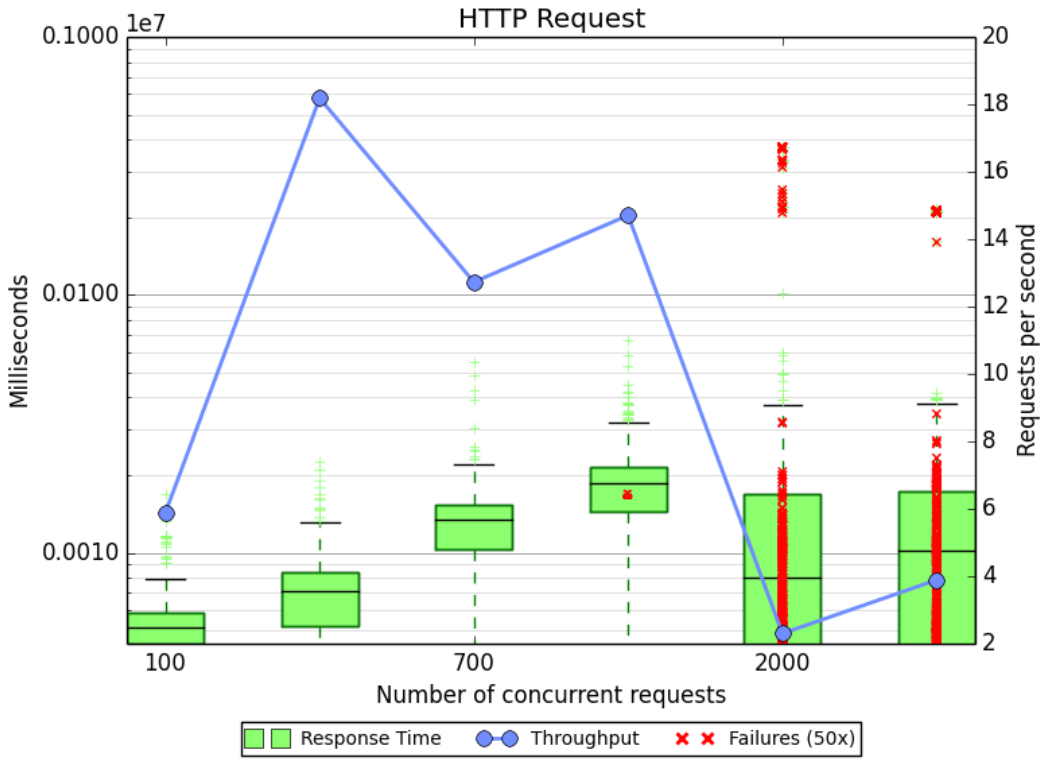

Figura 12: Diagrama com exemplos de requisições a funcionalidade de baixar sinais.

A partir da Tabela 10 e do gráfico da figura 12, é possível observar que para a máquina utilizada no teste o serviço foi capaz de atender até 700 requisições simultaneamente sem a ocorrência de falhas. Em que uma falha é caracterizada como uma requisição não concluída, seja com uma resposta inválida ou sem nenhuma resposta. A partir de 1000 requisições simultâneas o número de falhas apresentada ainda era baixo, com uma taxa média de 0,27% falhas no total. Entretanto, a partir de 2000 requisições simultâneas, o serviço começou a apresentar taxas mais altas de erro. É possível observar também que, a taxa de transferência diminuiu conforme o aumento do número de requisições simultâneas, e o tempo médio de resposta aumentou. Pode-se observar que houve um aumento considerável no tempo de resposta de acordo com o aumento do número de requisições. De acordo com a Tabela 8, houve um aumento de 95875ms quando se aumentou o número de requisições 1000 para 4000 requisições, o que representa um aumento de 86,2% no tempo de resposta.

5.2.2 Consulta de versão do dicionário

Os testes sobre a funcionalidade de consultar a versão do dicionário foram realizados utilizando a mesma metodologia dos testes apresentados na Seção 5.2.1. Várias requisições para consultar a versão do dicionário foram realizadas simultaneamente e a taxa de erro, taxa

de transferência e tempo médio de resposta foram mensuradas. Na Tabela 12 e na Figura 13 são apresentados esses resultados.

Tabela 12: Informações do teste de carga e desempenho da funcionalidade de consultar versão do dicionário.

Número de Requisições	**Taxa de falha**	**Taxa de transferência**	**Tempo médio de resposta**
1	0%	986,98kb/sec	1467ms
100	0%	965,98kb/sec	1342ms
400	0%	900,15kb/sec	1315ms
700	0%	890,21kb/sec	1389ms
1000	0%	889,48kb/sec	1412ms
2000	28,38%	556,79kb/sec	1512ms
4000	42,61%	400,21kb/sec	1742ms

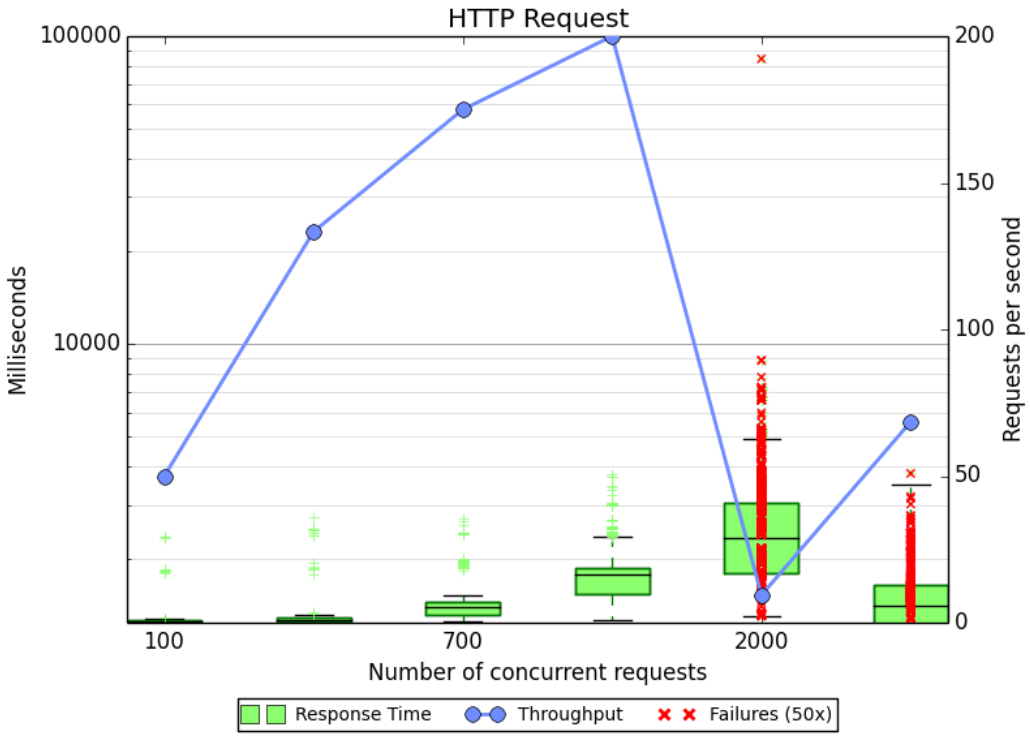

Figura 13: Diagrama com exemplos de requisições a funcionalidade de consultar versão do dicionário.

De acordo com a Tabela 11 e o gráfico da Figura 13, é possível observar que a máquina utilizada nos testes foi capaz de atender até 1000 requisições simultâneas sem a ocorrência de falhas. Em que uma falha é caracterizada como uma requisição não concluída, seja com uma

resposta inválida ou sem nenhuma resposta. A partir de 2000 requisições simultâneas o serviço começou a apresentou taxas de erro superiores a 25%. Além disso, também foi possível observar que a taxa de transferência diminuiu conforme o aumento do número de requisições simultâneas e o tempo médio de resposta aumentou. Podemos observar que o tempo de resposta oscilou, aumentando na maioria dos casos. Porém, não é possível afirmar que o aumento foi causado pelo aumento no número de requisições, pois, de acordo com a Tabela 9, verifica-se que em alguns casos o tempo de resposta chegou a diminuir com o aumento no número de requisições.

6 CONSIDERAÇÕES FINAIS

As novas tecnologias vêm quebrando barreiras e estimulando a inclusão das pessoas que possuem algum grau de deficiência auditiva na sociedade. Este trabalho propõe o desenvolvimento de um serviço para gestão e provimento distribuído de um dicionário de LIBRAS aberto para a Suíte VLIBRAS. Além disso, por se tratar de um dicionário aberto, a solução também prevê sua utilização por aplicações de terceiros que envolvam dicionários de LIBRAS e outros tradutores automáticos para LIBRAS. Para o desenvolvimento do serviço proposto no dicionário, foi analisada a necessidade de funcionalidades do serviço com a equipe de desenvolvedores do VLIBRAS, bem como do levantamento das necessidades para o desenvolvimento de novas aplicações.

Para avaliar a solução, foi realizado um teste integrando as algumas das funcionalidades do serviço nas aplicações da suíte de ferramentas de tradução automática VLIBRAS. Para isso, inicialmente se montou uma arquitetura da integração das ferramentas com o serviços, onde se definiu as responsabilidades inerentes ao modelo conceitual da solução de cada um. Após a definição da arquitetura, realizou-se a integração com as ferramentas, sendo possível verificar se os requisitos levantados para o serviço atenderam as necessidades das ferramentas e validar a sua eficiência diante o objetivo de ser um serviço para controle de dicionário de LIBRAS em várias ferramentas.

Devido a integração com alguma das aplicações da Suíte VLIBRAS, é possível observar que os principais objetivos foram alcançados, conseguimos idealizar e desenvolver o serviço dentro das necessidades levantadas para o desenvolvimento e atender a demanda de novas e atuais aplicações com acessibilidade para surdos.

6.1 Sugestões de Trabalhos Futuros

Como proposta de trabalhos futuros pretendemos:

- *Realizar testes com outros sistemas de tradução automática* e em outras plataformas utilizando outras funcionalidades do sistema que não foram testadas devido a falta de meios para isso;
- *Realizar testes de desempenho, de carga entre outros* mais exaustivos com um maior número de usuários em uma infraestrutura com mais servidores e com outras

funcionalidades do serviço, além de verificar o comportamento dos servidores utilizados para os testes para verificar a utilização dos recursos;

- *Disponibilizar um material* explicativo para que a comunidade possa começar atualizar nosso serviço;
- *Analisar os relatórios de uso das funcionalidades* para validar melhor os resultados e buscar melhorar o serviço dentro das necessidades que podem surgir para outros possíveis usuários a partir da análise do seu funcionamento.

REFERÊNCIAS BIBLIOGRÁFICAS

RUSSELL, S.; NORVIG, P. **Inteligência artificial**. [S.l.]: CAMPUS - RJ, 2004. ISBN 9788535211771.

ARAÚJO, T. M. U. de. **Uma Solução para Geração Automática de Trilhas em Língua Brasileira de Sinais em Conteúdos Multimídia**. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, Natal, Brasil, 2012. 9, 10, 13, 37

DOMINGUES, L. de A (2013). **CineLIBRAS: Uma Proposta para Geração Automática e Distribuição de Janelas em LIBRAS em Salas de Cinema**. Monografia (Graduação) – Universidade Federal da Paraíba, Rio Tinto – PB.

JANUÁRIO, G.; KOGA, M. L.; LEITE, L. POLI-LIBRAS: **Um Tradutor de Português para Libras**. 2010. Disponível em: http://www.polilibras.com.br/documentos/monografia.pdf?attredirects=0&d=1.

SILVA, D. A. N. dos S. (2012), **Uma Linguagem Expansível para Descrição da Língua Brasileira de Sinais**. Dissertação (Mestrado em Informática) - Universidade Federal da Paraiba - UFPB, João Pessoa - PB.

LIMA, M. A. C. B. (2015). **Uma proposta de Linguagem Expasível para Adequação Sintático-Semântica em Tradutores Automáticos para Libras**. Dissertação (Mestrado em Informática). Universidade Federal da Paraíba – UFPB, João Pessoa – PB.

DGE, D. de G. E. **Modelo de Acessibilidade em Governo Eletrônico. Documento de Referência**: Versão 3.0. 2011. Disponível em: http://emag.governoeletronico.gov.br/emag/emag-3.pdf.

LIRA, G. A. **O Impacto da Tecnologia na Educação e Inclusão Social da Pessoa Portadora de Deficiência Auditiva: Tlibras Tradutor Digital Português x Língua Brasileira de Sinais-Libras.** Boletim Técnico Senac, 2003. Disponível em: . Acesso em: 05/12/2010.

BUTTUSSI, F.; CHITTARO, L.; COPPO, M. **Using web3d technologies for visualization and search of signs in an international sign language dictionary**. In: Proceedings of the twelfth international conference on 3D Web technology. New York, NY, USA: ACM, 2007. (Web3D '07), p. 61–70. ISBN 978-1-59593-652-3. Disponível em: http://doi.acm.org/10.1145/1229390.1229401. 9

OLIVEIRA, Mauro, et al. PIMENTER: **O Computador 24 horas! Relatório Interno de Projeto de Extensão**. CEFETCE. Setembro 2006

IBGE. Instituto Brasileiro de Geografia e Estatística. 16 de Novembro de 2000.
IBGE. "Instituto Brasileiro de Geografia e Estatística." **Censo Demográfico 2000**. http://www.ibge.gov.br/home/presidencia/noticias/27062003censo.shtm. 16 de Novembro de 2011.

AVRAM, Abel. **Ruby on Rails vs. Node.js at LinkedIn**. Disponível em: http://www.infoq.com/news/2012/10/Ruby-on-Rails-Node-js-LinkedIn.