

# Uso de Encapsulamento Flexível em Tradutores Automáticos Português-LIBRAS para Aplicação em Cenários com Servidores Espelho Centralizados, Distribuídos e Híbridos

Francielly Grigório<sup>1</sup>, Daniel Ramos<sup>1</sup>, Rostand Costa<sup>1</sup>, Gutenberg Neto<sup>1</sup>, Alexandre Duarte<sup>1</sup>

<sup>1</sup>Laboratório de Aplicações de Vídeo Digital - LAVID  
Centro de Informática - UFPB  
Campus V - Mangabeira – João Pessoa – PB – Brazil – 58.055-000

**Abstract.** *In this paper we present the concept of a VLibras-Box, a building box for distributed sign-language translation infrastructures. A VLibras-Box encapsulates a Portuguese to Brazilian Sign Language translation service and additional components needed to combine multiple VLibras-Boxes in order to assemble a fault-tolerant and scalable translation infrastructure. We present three different application scenarios, hanging from a single VLibras-Box, where the translation endpoint is defined in an static way to multiple translation servers scenarios with dynamic selection and a two-way load balancing strategy to compose public, private and hybrid fault-tolerant distributed translation infrastructures.*

**Resumo.** *Neste artigo nós apresentamos o conceito de uma VLibras-Box, uma unidade para construção de infra-estruturas distribuídas para um serviço de tradução de linguagem de sinais. A VLibras-Box encapsula um serviço de tradução da língua brasileira de sinais e componentes adicionais necessários para combinar múltiplas VLibras-Boxes de modo a construir infra-estruturas escaláveis e tolerantes a falha. Nós apresentamos três cenários de aplicação diferentes, que variam de uma única VLibras-Box sendo utilizada, no qual o terminal de tradução é definido de maneira estática, a cenários com múltiplos servidores de tradução com seleção dinâmica e uma estratégia de balanceamento de carga duplo a fim de compor infra-estruturas distribuídas de tradução tolerantes a falha de caráter público, privado e híbrido.*

## 1. Introdução

Segundo o censo demográfico realizado em 2010 pelo Instituto Brasileiro de Geografia e Estatística (IBGE), cerca de 9,7 milhões de brasileiros possuem deficiência auditiva, o que representa 5,1% da população. Deste total, cerca de 2 milhões possuem a deficiência auditiva severa - 1,7 milhões têm grande dificuldade para ouvir e 344,2 mil são surdos - e 7,5 milhões apresentam alguma dificuldade auditiva [8]. Em termos mundiais, a estimativa da Organização Mundial de Saúde é de que aproximadamente 360 milhões de pessoas apresentem algum nível de deficiência auditiva [15]. Essa significativa parcela da população brasileira e mundial enfrenta diversos desafios no acesso à informação.

O principal problema é que a língua que o indivíduo usa para se comunicar depende de sua natureza além do grupo de indivíduos com o qual ele convive. Os ouvintes, por exemplo, comunicam-se por intermédio de línguas oralizadas, ou seja, através de

sons articulados que são percebidos pelo sistema auditivo. Já os surdos, por outro lado, encontraram na linguagem gestual-corporal um meio eficaz de comunicação como alternativa à falta de capacidade auditiva. Essa modalidade, denominada língua de sinais (LS), envolve elementos linguísticos manuais, corporais e faciais para articular os sinais que são compreendidos através do sistema visual. Portanto, a língua na qual o surdo consegue perceber e produzir de maneira natural é a LS, ao passo que as línguas orais, utilizadas cotidianamente pela maioria das pessoas e em praticamente todos os meios de comunicação, representam apenas “uma segunda língua”[7].

Não existe uma língua de sinais única, praticada em todo o mundo. Cada LS é uma língua natural, com léxico e gramática próprios, desenvolvida por cada comunidade de surdos ao longo do tempo, assim como cada comunidade de ouvintes desenvolveu a sua língua oral. É normal que cada país tenha a sua própria língua gestual (às vezes até mais de uma). No Brasil, por exemplo, tem-se a Língua Brasileira de Sinais (LIBRAS), enquanto que em Portugal existe a Língua Gestual Portuguesa (LGP).

Nos diferentes contextos da sociedade atual brasileira, incluindo os contextos das TICs em conjunto com a Internet, a informação é transmitida normalmente através da língua portuguesa escrita ou falada. Neste sentido, o nível de proficiência dos surdos na sua “segunda língua” pode tornar a leitura uma tarefa árdua e limitada [5], fazendo deste fator uma barreira a mais na inclusão digital. Os resultados de um estudo realizado por Wauters [16], tendo como público-alvo crianças e adolescentes surdos holandeses de 7 a 20 anos de idade, relata que apenas 25% deles possuem capacidade de leitura igual ou superior ao de uma criança sem deficiência de 9 anos. Portanto, se o texto for extremamente simples e claro, essa seria uma alternativa razoável para os surdos, mas não a única e, tampouco, a ideal.

Uma alternativa para tratar isso seria a tradução do conteúdo em língua portuguesa para LIBRAS. Quando se trata do acesso à conteúdo online, é necessário considerar fazer a tradução para LIBRAS não somente de conteúdo em áudio ou em vídeo, mas também de textos mais complexos dos sites. Essa já é uma tarefa hercúlea *per si* pelo volume envolvido e que também embute alguns desafios adicionais:

- *alto custo do serviço*: em 2013, a tradução de 60 minutos de um áudio em língua portuguesa para LIBRAS feita por um intérpretes humano custava R\$ 585,00 [9];
- *grande dinamismo de conteúdos da Internet*: a frequência com que novas páginas são inseridas na Web é intensa, assim como é a frequência de atualização dos conteúdos das mesmas.

Apenas considerando o conteúdo disponibilizado pelos sistemas de vídeo sob demanda (ou *VoD*, do inglês *Video on Demand*), fica bem evidente como prover acessibilidade para pessoas surdas é uma missão de alta complexidade. O YouTube, por exemplo, optou por tornar seus vídeos acessíveis para surdos através da disponibilização de legendas. Essa alternativa pode ter sido escolhida devido ao alto custo da tradução de seus vídeos para todas as línguas de sinais existentes por intérpretes humanos. São milhões de usuários em praticamente todos os países do mundo e uma enormidade de vídeos existentes no acervo, o qual cresce continuamente. Os gastos seriam extremamente elevados, e possivelmente não haveriam intérpretes suficientes para a demanda.

Neste contexto, uma das abordagens mais promissoras para tais cenários é a

utilização de aplicações para tradução automática da língua oral para a língua de sinais. No caso do Brasil, atualmente existem pelo menos quatro alternativas disponíveis para tradução da língua portuguesa para LIBRAS: *HandTalk*[1], *ProDeaf*[2], *Rybená*[3] e *VLIBRAS*[4, 6].

Dentre as soluções listadas, a única gratuita e de código aberto é a *Suíte VLIBRAS*, a qual é a base deste trabalho. Uma motivação adicional para sua utilização advém do fato dela apresentar uma arquitetura flexível, que permite gerar a tradução para diferentes tipos de conteúdos digitais, como texto, áudio e vídeo (legenda, *closed caption* e áudio) em diferentes plataformas (windows, linux, osx, *plugins* para navegadores, tradutores *online*, TV Digital etc).

Essa grande variedade de formas e plataformas pela qual se dá a oferta de serviços de tradução da Suíte VLibras requer que a uma organização da pilha interna de componentes seja flexível e modular, podendo ser combinada de várias formas. Uma das estratégias usadas foi o isolamento dos mecanismos de tradução propriamente dita das interfaces de ativação, usando *web services* para o processamento de retaguarda. Além disso, a inviabilidade técnica da tradução de conteúdo em larga escala de forma client-side utilizando o VLIBRAS se deve a complexidade do sistema como um todo, o que iria demandar um processamento excessivo e muitas vezes inviável no lado do cliente.

Atualmente, tal processamento de retaguarda do VLIBRAS é baseado em uma arquitetura tradicional, sendo operacionalizada por infraestruturas computacionais centralizadas, virtualizadas ou não. Contudo, sabe-se que sistemas centralizados apresentam várias desvantagens quando comparados a sistemas similares com abordagem distribuída. A principal delas é a impossibilidade de adoção de estratégias mais colaborativas para alocar recursos para o serviço, tais como a disponibilização de servidores adicionais, requerendo normalmente a figura de patrocinadores formais.

Com o objetivo de endereçar os aspectos descritos anteriormente e ampliar as possibilidades para montagem de um serviço de larga escala para tradução automática de conteúdo digital em Português para LIBRAS, o presente trabalho investiga uma abordagem de encapsulamento da *Suíte VLibras* para facilitar a sua implantação. Assim, o encapsulamento de um sistema de tradução de Português para LIBRAS em uma estrutura, denominada *VLibras-Box*, que possa ser facilmente utilizada para compor diferentes plataformas distribuídas de tradução é o principal foco da abordagem descrita no presente trabalho. A arquitetura da *VLibras-Box* foi modelada de modo a permitir que cada unidade de processamento seja facilmente configurável e gerível e que essas unidades possam ser instanciadas a medida que os recursos sejam solicitados. Assim, dadas as características da *VLibras-Box*, é viável a construção de redes distribuídas e colaborativas, na qual vários domínios administrativos possam compartilhar recursos a serem utilizados por diferentes usuários. A ideia é permitir o provimento de infraestruturas de suporte para um serviço nacional de tradução LIBRAS que seja capaz de utilizar recursos computacionais centralizados, distribuídos ou híbridos de forma transparente, escalável e tolerante a falhas.

O restante do documento está organizado da seguinte forma: a Seção 2 apresenta o conceito de *VLibras-Box* e os principais aspectos de sua arquitetura; a Seção 3 aborda os cenários de aplicação da abordagem proposta; a Seção 4 detalha aspectos do *Service*

*Discovery* adotado para viabilizar os cenários previstos; a Seção 5 analisa os resultados obtidos; a Seção 6 exhibe alguns trabalhos relacionados com a nossa pesquisa e, finalmente, a Seção 7 traz as considerações finais e alguns dos passos futuros da nossa pesquisa.

## 2. VLibras-Box: Conceito e Arquitetura

A *VLibras-Box* foi projetada para funcionar como uma instância independente que incorpora todas as pilhas de serviços, devidamente orquestradas, necessárias para o processamento exigido para que uma requisição de tradução seja atendida. Cada *VLibras-Box* representa um servidor espelho utilizado para compor uma infra-estrutura de servidores redundante e tolerante a falhas. Esses espelhos possuem duplicações dos processos e transações do servidor primário.

A pilha de serviços que compõe uma *VLibras-Box* conta não só com a suite de software responsável pelo processo de tradução propriamente dito, mas também com componentes de um *Serviço de Descobrimto*. Soluções de *Service Discovery*, desenvolvidas especificamente para composição da plataformas distribuídas e tolerantes a falhas, contam com três elementos principais: um diretório de serviço consistente e de alta disponibilidade, que serve como um registro central; um mecanismo para registro e monitoramento da saúde dos serviços; e um mecanismo para localizar e conectar serviços.

O *registro de serviço* é o processo de registro da localização de cada serviço em um registro central (*diretório de serviços*). Geralmente é registrado o endereço do serviço e suas credenciais de autenticação, assim como detalhes do ambiente no qual ele opera. Por outro lado, a *descoberta de serviço* é o processo de consulta do registro central feito pelo cliente, de modo a conhecer a localização dos serviços aptos. Na abordagem descrita, cada instância de uma *VLibras-Box* representa o serviço disponibilizado.

### 2.1. Arquitetura de uma VLibras-Box

Nesta seção descrevemos a arquitetura de uma *VLibras-Box*, exibida na Figura 1, assim como cada um de seus componentes internos.

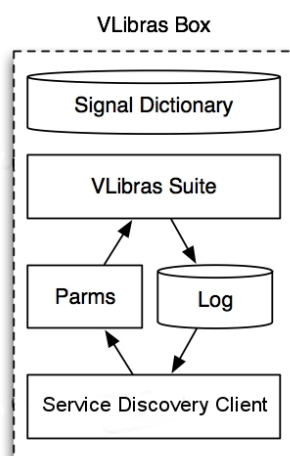


Figura 1. Componentes da VLibras-Box.

- **Dicionário de Sinais de LIBRAS.** O *Signal Dictionary* é um espelho local do repositório de sinais em LIBRAS utilizados para geração dos vídeos traduzidos.

Esse espelho local é atualizado automaticamente sempre que novos sinais são gerados e registrados no repositório central de sinais. O repositório armazena também as regras para concatenação dos sinais de forma a possibilitar a geração de vídeos com transições mais suaves.

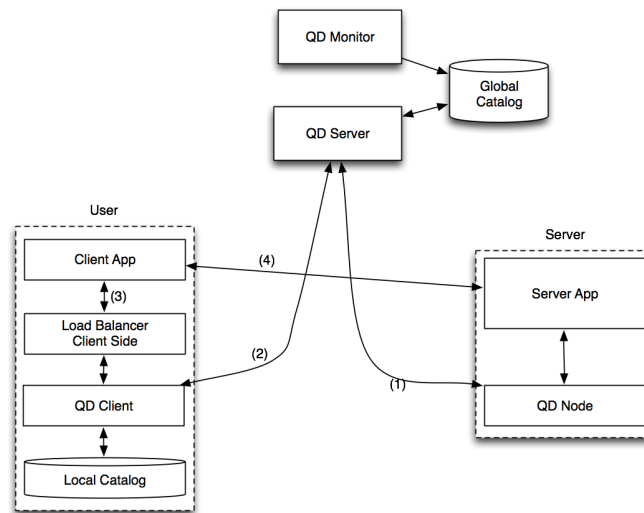
- **VLibras-Suite.** A *VLibras-Suite* [4, 6] é o motor responsável por todas as etapas do processo de tradução de texto escrito em Português para vídeo descrevendo o mesmo conteúdo em LIBRAS. Essa suite reúne uma coleção de aplicações que se destinam a realizar cada um dos passos necessários para completar uma tradução Português-LIBRAS.
- **Cliente para Serviço de Descobrimto.** Uma parte muito importante da *VLibras-Box* é a inclusão de um cliente para algum serviço de descobrimto. Este é um componente fundamental para a criação de infraestruturas de tradução flexíveis. Na Figura 1 este componente é denominado *Service Discovery Client*. O *Service Discovery Client* é o componente responsável por cadastrar e manter um espelho na infraestrutura de servidores. É esse componente que encapsula a comunicação entre a *VLibras-Box* e o *Service Discovery Server*, um servidor principal responsável pelo registro dos diferentes espelhos disponíveis na infraestrutura distribuída. O servidor principal mantém o registro de todas as *VLibras-Box* ativas na rede, responsável por disponibilizar aos clientes um catálogo listando todas as *VLibras-Box* aptas a receber pedidos no momento da requisição. A comunicação entre o *Service Discovery Client* e a *VLibras Suite* se dá através de arquivos de configuração, onde são armazenados os parâmetros de configuração do sistema de tradução e os arquivos de log, utilizados para coleta de dados referentes ao funcionamento do sistema, propagados posteriormente para atualização do cadastro central mantido pelo *Service Discovery Server*.

Na Seção 3, detalhamos o funcionamento do *QD Framework*, um arcabouço utilizado para composição de infraestruturas distribuídas e tolerantes a falhas para servidores espelho [14]. Nesse framework, o *QD Node* apresenta a função do *Service Discovery Client* e o *QD Server* por sua vez, desempenha o papel do *Service Discovery Server*.

## 2.2. Configuração de uma VLibras-Box

O funcionamento de uma *VLibras-Box*, e por conseguinte, do serviço de tradução por ela encapsulado, pode ser customizado através de uma série de parâmetros de configuração.

- **Modos de operação.** No modo de operação bloqueante, após efetuar uma requisição o cliente permanece a espera da resposta do servidor, deixando o canal de comunicação entre ambos aberto até que uma resposta seja recebida. Por outro lado, no modo de operação não-bloqueante, o sistema retorna imediatamente para a aplicação, e o cliente recebe um identificador para que possa checar o estado da tradução e posteriormente requisitar o vídeo gerado pelo tradutor.
- **Política de Armazenamento.** A *VLibras-Box* pode ser configurada para utilizar tanto armazenamento interno quanto armazenamento externo para salvar os vídeos gerados por seu tradutor.
- **Endereço do Service Discovery Server.** De modo a se tornar visível na infraestrutura distribuída de tradução, cada *VLibras-Box* deve se registrar em um *Service Discovery Server*, o servidor principal que mantém o registro de todas as *VLibras-Box* ativas na rede, e é o responsável por disponibilizar aos clientes um catálogo listando todas as *VLibras-Box* aptas a receber pedidos no momento da requisição.



**Figura 2. Arquitetura do QD Framework**

### 3. Registro Dinâmico de VLibras-Box e sua Descoberta pelos Clientes

Esta seção apresenta de forma resumida a arquitetura do *QD Framework* focando nos serviços necessários para permitir o registro de cada *VLibras-Boxes* em uma rede de tradução e viabilizar a sua descoberta e utilização pelos clientes.

O *QD Framework* endereça o problema da localização de recursos em uma rede distribuída ao ser uma solução projetada especificamente para a descoberta e registro dinâmico de serviços, denominada *Service Discovery*. Soluções de *Service Discovery* contam com três elementos principais: um diretório de serviço consistente e de alta disponibilidade, que serve como um registro central; um mecanismo para registro e monitoramento da saúde dos serviços; e um mecanismo para localizar e conectar serviços.

O *registro de serviço* é o processo de registro da localização de cada serviço em um registro central (*diretório de serviços*). Geralmente é registrado o endereço do serviço e suas credenciais de autenticação, assim como detalhes do ambiente no qual ele opera. Por outro lado, a *descoberta de serviço* é o processo de consulta do registro central feito pelo cliente, de modo a conhecer a localização dos serviços aptos.

O *QD framework* é um *Service Discovery* genérico dividido em 3 componentes principais, ilustrados na Figura 2, e descritos a seguir.

O *QD Server* é o componente responsável por gerenciar a lista de servidores espelho ativos pertencentes à malha, ou seja, o catálogo dinâmico de *VLibras-Boxes*. Além de gerenciar o repositório de tradutores ativos, o *QD Server* também possui a funcionalidade de organizar mini-catálogos personalizados dinamicamente, fornecendo para cada cliente uma lista contendo os servidores mais adequados de acordo com cada necessidade.

Quando uma lista de tradutores é requisitada por um cliente, o *QD Server* realiza uma triagem entre todos as *VLibras-Boxes* permitidas para o domínio específico do cliente e, a partir das informações cadastradas, retorna apenas os servidores considerados mais apropriados. Podem ser removidos tradutores que não tenham atualizado suas informações recentemente, que tenham uma alta taxa de falha nas últimas requisições ou

que estejam sobrecarregados, por exemplo.

Um requisito fundamental para a construção da malha de tradutores foi a incorporação de um módulo *QD Node* a cada *VLibras-Box*. Como ele é o componente responsável do *QD Framework* pela comunicação de cada servidor espelho com o *QD Server*, ele permite que o estado da rede de tradutores disponíveis seja permanentemente mantido. Neste sentido, o *QD Node* desempenha duas funções principais:

- Efetuar o registro inicial da *VLibras-Box* junto ao *QD Server*, indicando a sua configuração, forma de acesso, entre outros parâmetros fundamentais;
- Enviar um *heartbeat* periódico para o *QD Server* indicando que continua ativo, de forma que o mesmo possa continuar sendo incluído nas listas dinâmicas enviadas para os clientes.

Cada cliente de tradução, por sua vez, precisa incorporar um *QD Client*. Esse é o componente do *QD Framework* responsável por requisitar uma lista de tradutores ao *QD Server* e, a partir dessa lista, selecionar o mais apropriado a ser utilizado em cada requisição.

O funcionamento do *QD Client* baseia-se em um tipo de balanceamento comum na computação em nuvem, denominado balanceamento de carga no lado do cliente (do inglês *client-side load balancing*) e foi projetado de forma a permitir a utilização de diversas heurísticas complementares para a escolha do tradutor mais adequado, inclusive considerando uma melhor distribuição da demanda por toda a malha.

O *QD Client* também é responsável por manter uma lista local de tradutores cada vez que recebe uma nova lista do *QD Server*. Isto tem duas razões: a primeira é garantir que, mesmo que o *QD Server* esteja inacessível por algum motivo, o cliente ainda possa fazer a escolha de um tradutor. A segunda razão é para evitar que o *QD Client* precise requisitar uma nova lista do *QD Server* a cada requisição. Isso diminui a carga no *QD Server* e minimiza o tempo de escolha do tradutor, posto que a atualização da lista local de tradutores pode ser realizada apenas periodicamente ou quando necessário.

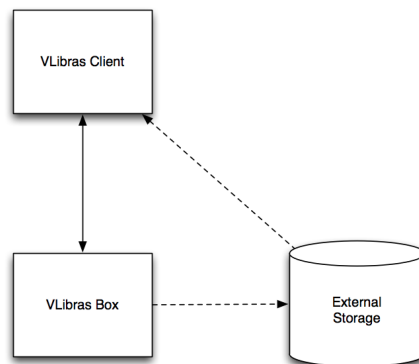
Quando requisita uma nova lista ao *QD Server*, o *QD Client* envia informações consolidadas sobre as requisições de tradução realizadas desde a última atualização, incluindo tradutores utilizados e tempo médio de resposta e os eventuais erros de comunicação. Com isto, o *QD Server* pode manter atualizadas as informações sobre o desempenho dos tradutores no catálogo dinâmico.

## 4. Cenários de Utilização

Nesta seção são descritos diferentes cenários de utilização para o conceito de *VLibras-Box* proposto.

### 4.1. Servidor Estático

Este cenário, ilustrado na Figura 3, representa o cenário de utilização mais simples, no qual uma única *VLibras-Box*, conhecida por todos os clientes, atende todo o sistema. Nesse tipo de arquitetura, os clientes se conectam diretamente a um único servidor que é encarregado de processar todas as requisições. O resultado da tradução pode ser armazenado tanto no sistema de arquivos da própria *VLibras-Box* quanto em um servidor de armazenamento externo.



**Figura 3. Interação direta entre um VLibras-Client e a VLibras-Box.**

A principal desvantagem deste cenário é que se o servidor falhar o serviço de tradução deixa de funcionar por completo. Assim, qualquer falha operacional, como falhas de hardware ou de software, causa uma interrupção no serviço de tradução. Além disso, este tipo de arquitetura apresenta um limite fixo para o número máximo de requisições simultâneas que podem ser atendidas, tornando-se instável ou mesmo indisponível quando a carga de trabalho atinge valores próximos a este limite.

## 4.2. Servidor Dinâmico

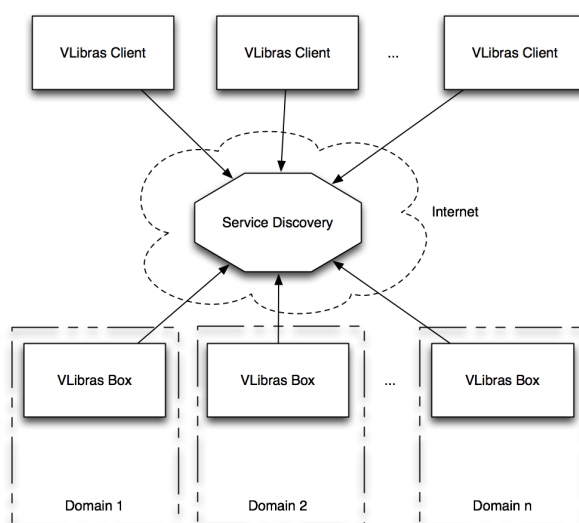
Em cenários com seleção dinâmica de servidores, o servidor a ser utilizado pelo cliente só é conhecido no momento da solicitação e pode ser alterado entre solicitações subsequentes. Esta alocação dinâmica nos permite montar infraestruturas distribuídas de forma muito mais eficiente e que podem tolerar falhas e escalar sua capacidade de atendimento de acordo com o aumento da carga de trabalho.

Nessa abordagem, a construção de infraestruturas para seleção dinâmica de servidores se dá através do processo de replicação das *VLibras-Box* e dos registros dessas diversas unidades de tradução em um sistema de descoberta de serviço. Assumimos para simplificação da explicação a utilização do *QD Framework* para registro e descoberta de *VLibras-Box* disponíveis em diferentes cenários dinâmicos, descritos a seguir.

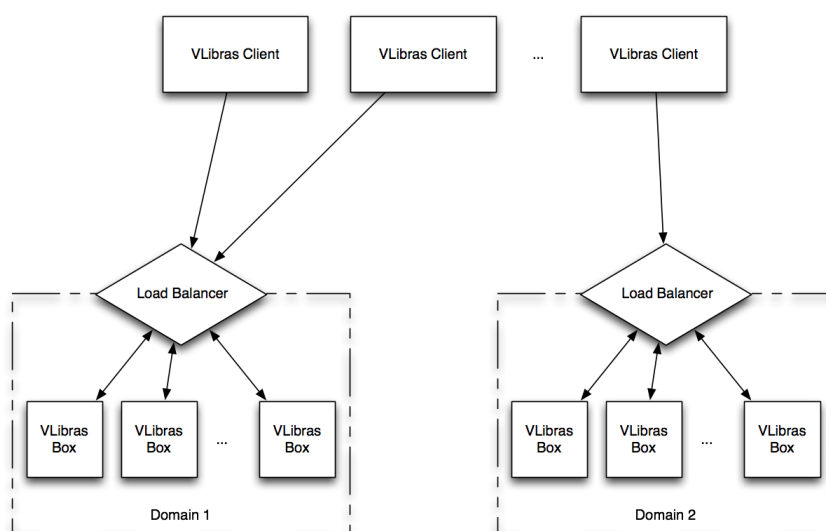
### 4.2.1. Infraestrutura Pública de Tradução

Neste cenário de utilização (Figura 4), diferentes domínios administrativos são responsáveis por disponibilizar *VLibras-Box* para serem utilizadas por qualquer usuário do sistema. Órgãos governamentais, empresas, e associações de surdos, por exemplo, poderiam manter servidores de tradução independentes que se registrariam em um serviço centralizado de cadastro. Usuários do sistema consultariam este serviço de cadastro, responsável por indicar o servidor mais apto a ser utilizado dado o tipo de tradução a ser realizada e a carga dos servidores de tradução naquele momento, antes do pedido de tradução ser encaminhado à *VLibras-Box* selecionada.





**Figura 4. Infraestrutura Pública de Tradução**

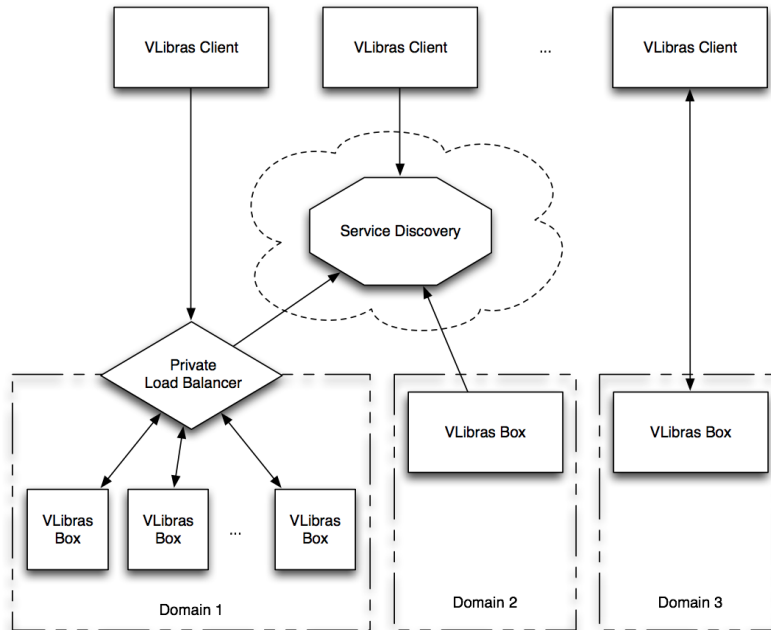


**Figura 5. Infraestrutura Privada de Tradução**

#### 4.2.2. Infraestrutura Privada de Tradução

Um segundo cenário dinâmico (Figura 5), prevê que cada domínio administrativo disponibilize uma ou mais *VLibras-Box* para atender exclusivamente os seus próprios usuários. Cada domínio poderia manter um conjunto de servidores de tradução para atender exclusivamente usuários de seus próprios sites.

Para ilustrar mais um aspecto da flexibilidade possível adicionamos ao cenário a utilização de balanceadores de carga (*Load Balancers*). Neste cenário, cada *VLibras-Box* após ser instanciada, se registra no componente balanceador. Estes balanceadores de carga devem apresentar a mesma interface que uma *VLibras-Box* e são responsáveis por direcionar as requisições recebidas para algum dos servidores de tradução disponíveis, de acordo com alguma heurística de balanceamento, além de descartar solicitações para



**Figura 6. Infraestrutura Híbrida de Tradução**

tradução que fujam ao escopo do domínio administrativo provedor do serviço de tradução.

### 4.3. Infraestrutura Híbrida de Tradução

Uma infraestrutura híbrida de tradução pode combinar todos os tipos de cenários descritos anteriormente, incluindo o cenário estático e os dois dinâmicos, para definir uma infraestrutura extremamente flexível como ilustrado na Figura 6.

## 5. Resultados e Análise

Foi realizada uma série de testes de validação de um protótipo de *VLibras-Net* baseada em uma estrutura distribuída real. Os componentes âncora de retaguarda, *QD Server* e *QD Monitor*, ficaram hospedados nos servidores do LAVID, em João Pessoa, e o *pool* de *VLibras-Boxes* foi instanciado em 32 nós de um cluster localizado no CCAE, na cidade de Rio Tinto. Os clientes simultâneos foram executados de outras localizações.

Uma rodada de medições de campo da **Suíte VLibras** nativa (sem estar encapsulada como uma **VLibras Box**) precedeu os testes de validação com o intuito de estabelecer um tempo médio de tradução em um ambiente controlado, o qual foi usado nos experimentos como valor de referência para aplicação de retardos (*delays*) controlados.

Os testes foram organizados em 4 grupos, cada um representando um experimento com uma estratégia distinta e uma finalidade específica:

- Experimento 1 - *Balanceamento de Carga em Ambiente Homogêneo*: Com o retardo de todos os *QD Nodes* configurados para o tempo de referência observado nos testes de campo, cada uma das heurísticas client-side foi executada por 15 clientes simultâneos em 5 servidores de tradução. As observações de interesse foram: tempo médio de resposta (TMR) (para indicar o *overhead* do *QD Framework*) e a quantidade de solicitações atendida por cada *QD Node* (eficácia da heurística de balanceamento em um ambiente homogêneo).

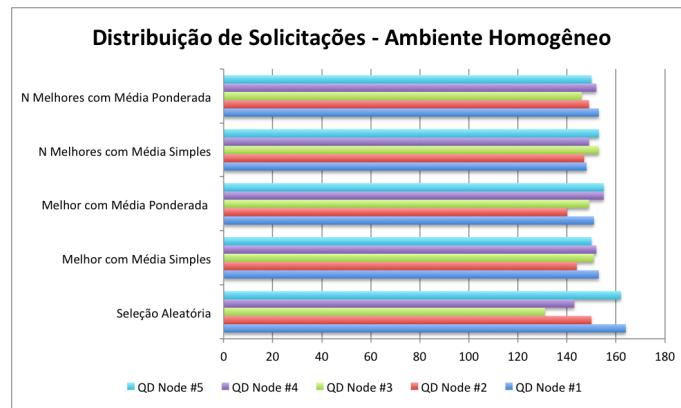


Figura 7. Distribuição de Solicitações em *Pool* Homogêneo

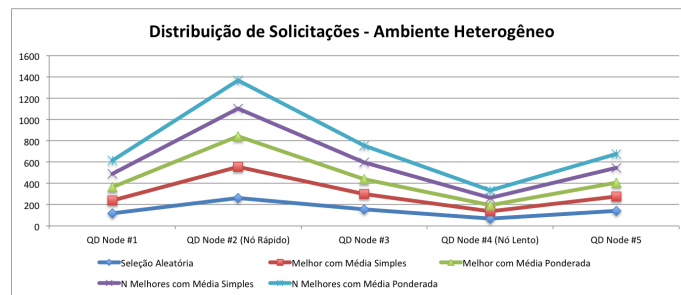
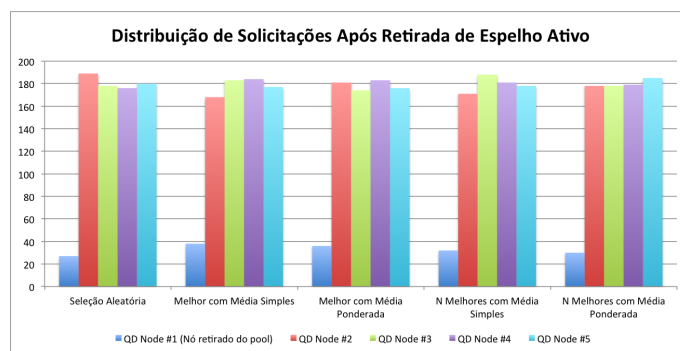


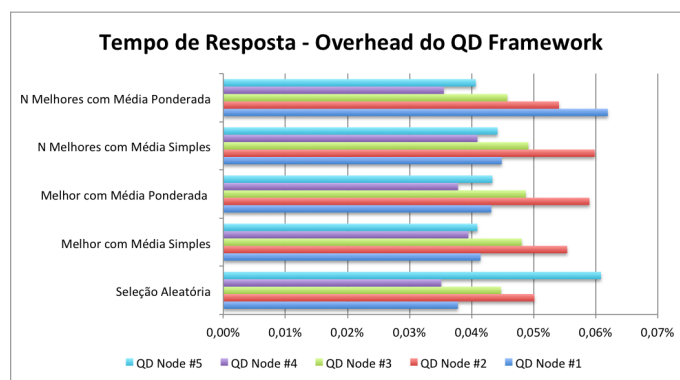
Figura 8. Distribuição de Solicitações em *Pool* Heterogêneo

- Experimento 2 - *Balanciamento de Carga em Ambiente Heterogêneo*: Semelhante ao Experimento 1, apenas a configuração dos retardos muda: metade dos servidores de tradução com o valor de referência, um com a metade (mais rápido) e o outro com um retardo artificial seis vezes maior (mais lento). O objetivo foi caracterizar um ambiente heterogêneo com máquinas sobrecarregadas e máquinas padrão ou com pouca carga. As observações de interesse foram as mesmas e o importante é verificar se o TMR fica normalizado e se os servidores com menor carga são priorizadas até ocorrer um equilíbrio.
- Experimento 3 - *Tolerância a Falhas*: Semelhante a configuração do Experimento 1 (ambiente homogêneo), o teste é iniciado com todos os *QD Nodes* ativos para cada heurística e um deles é desativado após um período de estabilização. O objetivo é verificar se há uma redistribuição adequada das solicitações pelos servidores restantes.
- Experimento 4 - *Elasticidade*: Semelhante a configuração do Experimento 1 (ambiente homogêneo), exceto que o *pool* de tradutores deve ser iniciado com um *QD Node* a menos, o qual será ativado após um período inicial de estabilização.

Considerando o cenário híbrido, mais complexo, foi possível observar uma grande sensibilidade e adaptação de todas as heurísticas do *QD Framework* a cada contexto. Em particular, a heurística "N Melhores com Média Ponderada" mereceu destaque, sendo a mais eficiente tanto em cenários sem contenção de recursos e equilíbrio de desempenho (Figura 7) quanto em ambientes com nós medianos, nós sobrecarregados e nós mais rápidos (Figura 8). Também foi dela a melhor eficácia de distribuição de solicitações na presença de falhas (Figura 9). Apesar de permitir uma excelente operacionalização de to-



**Figura 9. Distribuição de Solicitações Após Falha de Nó**



**Figura 10. Sobrecarga do QD Framework sobre o Tempo Médio de Resposta**

dos dos cenários previstos, o *QD Framework* apresentou um acréscimo mínimo no tempo de resposta, abaixo de 1% (Figura 10).

## 6. Trabalhos Relacionados

Recentemente, acadêmicos de diversas áreas têm investigado maneiras de combinar diferentes tecnologias para suportar dinamicamente serviços configuráveis. Alguns problemas envolvidos nesse meio são: detecções de falhas e reações do sistema a elas, medições de métricas de QoS, tais como taxas de perda e taxa de transferência per-link, implantação e terminação de aplicativos, depuração e monitoramento, virtualização de nós distribuídos, entre outros. Abaixo são apresentados estudos que, assim como o presente trabalho, também visam oferecer soluções para os problemas listados acima. O conceito de VLibras-Box se assemelha ao conjunto de trabalhos citados abaixo, ao propor uma maneira de gerar infraestruturas distribuídas compostas por blocos autônomos configuráveis.

O iOverlay [12], uma infraestrutura de middleware leve e de alto desempenho, foi proposto com o objetivo de solucionar os entraves encontrados no momento de oferecer um sistema distribuído configurável de uma maneira nova, fornecendo camadas limpas e bem documentadas de componentes de middleware. Outra solução similar é o Accord [13], um framework de programação baseada em componentes para apoiar o desenvolvimento de aplicações autônomas de auto-desenvolvimento. A solução permite não só o desenvolvimento de componentes autônomos distribuídos mas também a formulação de aplicações autônomas completas. Similarmente ao Accord, o middleware Gridkit [11]

expande a arquitetura orientada a serviços básicos para lidar com algumas deficiências, tais quais a falta de suporte a rica diversidade de 'tipos de interação' de comunicação que são exigidos por aplicações mais avançadas (por exemplo, media streaming, interação peer-to-peer, publish-subscribe).

Por fim, o CANS [10] se apresenta como uma abordagem automática para a implantação dinâmica de componentes intermediários que podem ser reconfigurados de forma eficiente em tempo de execução, de forma a permitir o acesso on-line a serviços de Internet. Sua infraestrutura é baseado em três mecanismos-chave: (a) uma especificação de alto nível integrada baseada em tipo de componentes e recursos de rede; (b) uma estratégia de criação de caminho automático; e (c) suporte do sistema para reconfiguração de caminho de baixo-overhead.

## 7. Conclusão

O presente trabalho introduz o conceito de *VLibras-Box*, um bloco conceitual utilizado para composição de infraestruturas distribuídas para tradução de conteúdo multimídia em Português para LIBRAS.

Uma *VLibras-Box* encapsula um sistema de tradução de Português para LIBRAS, denominado **Suíte VLIBRAS**, e componentes adicionais necessários para combinar múltiplas *VLibras-Boxes* de forma a montar uma infraestrutura de tradução escalável e tolerante a falha.

Apresentamos três diferentes cenários de aplicação, variando desde um cenário mais simples, composto por uma única *VLibras-Box* responsável por atender todas as requisições dos clientes e selecionada estaticamente até cenários mais interessantes, onde múltiplas *VLibras-Boxes* são combinadas com serviços de descobrimento e de balanceamento de carga para montar um cenário onde o servidor de tradução a ser utilizado para atender uma determinada requisição é selecionado de forma dinâmica, segundo uma estratégia de balanceamento de carga em dois estágios. Nesses tipos mais robusto de cenários também leva-se em consideração diferentes aspectos da requisição como tipo de documento a ser traduzido, origem do documento e até mesmo a origem do cliente.

Estes cenários evidenciam a viabilidade da implantação de infraestruturas públicas, privadas ou híbridas para o fornecimento de um serviço de tradução de conteúdo multimídia em Português para LIBRAS, representando um importante passo em direção à inclusão de pessoas surdas no atual cenário das tecnologias da comunicação e informação.

Como trabalhos futuros pretendemos avaliar diferentes heurísticas para nossa estratégia de escalonamento em dois níveis e implantar uma infraestrutura piloto para tradução de conteúdo em Português para LIBRAS no contexto de um projeto financiado pelo Ministério do Planejamento.

## Referências

- [1] Hand Talk. <http://www.handtalk.me/>. Accessed 25-Jun-2015.
- [2] ProDeaf. <http://prodeaf.net/OQueE/>. Accessed 25-Jun-2015.
- [3] Rybená Web. <http://www.rybena.com.br/site-rybena/conheca-o-rybena/web>. Accessed 25-Jun-2015.

- [4] Tiago Maritan Ugulino de Araújo. Uma solução para geração automática de trilhas em língua brasileira de sinais em conteúdos multimídia. 2012.
- [5] Rachel Colacique Gomes. E-acessibilidade para surdos. *Revista Brasileira de Tradução Visual*, 7(7), 2011.
- [6] Tiago Maritan U De Araújo, Felipe LS Ferreira, et al. An approach to generate and embed sign language video tracks into multimedia contents. *Information Sciences*, 281:762–780, 2014.
- [7] Maria Cecília Rafael de Góes. *Linguagem, surdez e educação*. Autores Associados, 2002.
- [8] IBGE Censo Demográfico. Instituto brasileiro de geografia e estatística, 2012, 2010.
- [9] SINTRA Sindicato Nacional dos Tradutores. Valores de referência praticados a partir de janeiro de 2013. <http://www.sintra.org.br/site/?p=c&pag=precos.%204>, 2013.
- [10] Xiaodong Fu and Vijay Karamcheti. Automatic creation and reconfiguration of network-aware service access paths. *Comput. Commun.*, 28(6):591–608, April 2005.
- [11] Paul Grace, Geoff Coulson, Gordon Blair, Laurent Mathy, Wai Kit Yeung, Wei Cai, David Duce, and Chris Cooper. Gridkit: Pluggable overlay networks for grid computing. In *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, pages 1463–1481. Springer, 2004.
- [12] Baochun Li, Jiang Guo, and Mea Wang. ioverlay: A lightweight middleware infrastructure for overlay application implementations. In *Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware*, Middleware '04, pages 135–154, 2004.
- [13] Hua Liu. A component-based programming model for autonomic applications. In *Proceedings of the First International Conference on Autonomic Computing*, ICAC '04, pages 10–17, Washington, DC, USA, 2004. IEEE Computer Society.
- [14] Author names removed for double blind review. Qd framework: Usando catálogos dinâmicos para organizar servidores espelho em malhas de serviços escaláveis e resilientes. In *Submitted for the Proceedings of the 21th Brazilian Symposium on Multimedia and the Web*. ACM, 2015.
- [15] World Health Organization et al. Who: Deafness and hearing loss, fact sheet n 300 updated february 2013, 2013.
- [16] Loes Nel Wauters. *Reading comprehension in deaf children: The impact of the mode of acquisition of word meanings*. Nij // jmegem: EAC, Research Centre on Atypical Communication, Radboud University, 2005.