

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET

MARCELO LOPES DA SILVA

**USO DE REDES SOCIAIS COMO MECANISMO DE ACESSO À
COMPOSIÇÃO DE SERVIÇOS WEB NA UTFPR-CM**

TRABALHO DE CONCLUSÃO DE CURSO

CAMPO MOURÃO - PR

2013

MARCELO LOPES DA SILVA

**USO DE REDES SOCIAIS COMO MECANISMO DE ACESSO À
COMPOSIÇÃO DE SERVIÇOS WEB NA UTFPR-CM**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de Tecnólogo em Tecnologia em Sistemas para Internet.

Orientador: Prof. Me. Ivanilton Polato

CAMPO MOURÃO - PR

2013



ATA DA DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO

Às **vinte e uma horas** do dia **vinte e sete de setembro de dois mil e treze** foi realizada no Mini-auditório do EAD da UTFPR-CM a sessão pública da defesa do Trabalho de Conclusão do Curso Superior de Tecnologia em Sistemas para Internet do acadêmico **Marcelo Lopes da Silva** com o título **USO DE REDES SOCIAIS COMO MECANISMO DE ACESSO À COMPOSIÇÃO DE SERVIÇOS WEB NA UTFPR-CM**. Estavam presentes, além do acadêmico, os membros da banca examinadora composta pelo professor **Me. Ivanilton Polato** (Orientador-Presidente), pelo professor **Me. Luiz Arthur Feitosa dos Santos** e pelo professor **Me. Rodrigo Campiolo**. Inicialmente, o aluno fez a apresentação do seu trabalho, sendo, em seguida, arguido pela banca examinadora. Após as arguições, sem a presença do acadêmico, a banca examinadora o considerou **APROVADO** na disciplina de Trabalho de Conclusão de Curso e atribuiu, em consenso, a nota **8,5 (OITO E MEIO)**. Este resultado foi comunicado ao acadêmico e aos presentes na sessão pública. A banca examinadora também comunicou ao acadêmico que este resultado fica condicionado à entrega da versão final dentro dos padrões e da documentação exigida pela UTFPR ao professor Responsável do TCC no prazo de **onze dias**. Em seguida foi encerrada a sessão e, para constar, foi lavrada a presente Ata que segue assinada pelos membros da banca examinadora, após lida e considerada conforme.

Observações:

Campo Mourão, 27 de setembro de 2013.

A Folha de Aprovação assinada encontra-se na Coordenação do Curso

RESUMO

DA SILVA, Marcelo Lopes. USO DE REDES SOCIAIS COMO MECANISMO DE ACESSO À COMPOSIÇÃO DE SERVIÇOS WEB NA UTFPR-CM. 37 f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Campo Mourão - PR, 2013.

O advento dos serviços Web trouxe novas possibilidades para o uso de sistemas para Internet. A composição de serviços para a realização de tarefas mais complexas tem demonstrado ser uma maneira flexível e prática de construir novos sistemas a partir de serviços existentes. O uso dessas tecnologias no ambiente da UTFPR-CM é desejável para a promoção da integração entre sistemas e para evitar a sobrecarga dos sistemas existentes. Nesse contexto, o desenvolvimento de uma plataforma de serviços Web disponíveis aos acadêmicos e servidores é uma oportunidade de melhoria dos sistemas institucionais existentes. Neste trabalho, desenvolvemos uma rede de serviços Web, coordenados por um orquestrador de requisições, que realiza composições de serviços simples para a realização de tarefas mais complexas. Utilizando-se de um comunicador instantâneo como interface de uso, o sistema provê acesso aos serviços disponíveis de forma natural, simulando uma conversa com um ser humano. A abstração da complexidade proposta pelo sistema e sua flexibilidade de uso tornam esse trabalho uma solução de comunicação alternativa para os sistemas acadêmicos da UTFPR-CM.

Palavras-chave: Serviços Web, Redes Sociais, Composição, Orquestração, Chatbot, Bot

ABSTRACT

DA SILVA, Marcelo Lopes. . 37 f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Campo Mourão - PR, 2013.

The Web services advent has brought new possibilities of use for the internet systems. Web service composition to perform more complex tasks has proven to be a flexible and practical way to build new systems from existing services. The use of these technologies at the UTFPR-CM is desirable for the integration between existent systems and to avoid overload of such systems. In this context, the development of a Web services platform available to academic and faculty is an opportunity for improvement of existing institutional systems. In this work, we develop a network of Web services, coordinated by an orchestrator, performing compositions of simple services to perform more complex tasks. Using an instant messenger as user interface, the system provides access to the available services in a natural way by simulating a human conversation. The complexity abstraction proposed by the system and the ease of use make this solution a communication alternative for the UTFPR-CM systems.

Keywords: Web Services, Social Networks, Composition, Orchestration, Chatbot, Bot

LISTA DE FIGURAS

FIGURA 1	– Estrutura de comunicação de um serviço Web	8
FIGURA 2	– Coreografia de serviços Web	10
FIGURA 3	– Orquestração de serviços Web	10
FIGURA 4	– Exemplo de processo usando BPMN	12
FIGURA 5	– Tela do sistema RobôED	13
FIGURA 6	– Diagrama ilustrando o sistema proposto	16
FIGURA 7	– Diagrama de classes do orquestrador proposto	18
FIGURA 8	– Diagrama BPMN do serviço “Notas por Aluno”	19
FIGURA 9	– Diagrama BPMN do serviço “Pendências na Biblioteca”	19
FIGURA 10	– Diagrama BPMN do serviço “Consulta Horário de Aulas”	20
FIGURA 11	– Diagrama BPMN do serviço “Laboratórios Vagos”	20
FIGURA 12	– Diagrama BPMN do serviço “Reserva de Infraestrutura”	21
FIGURA 13	– Diagrama BPMN do serviço “Requerimento de Matrícula”	22
FIGURA 14	– Exemplo de comunicação entre usuário e o sistema proposto	24
FIGURA 15	– Código da implementação do teste de unidade para o Algoritmo Soundex	26
FIGURA 16	– Resultado do teste de unidade para o Algoritmo Soundex	27
FIGURA 17	– Diagrama de Classe que representa um Serviço Web	28
FIGURA 18	– Requisição XML solicitando perguntas	28
FIGURA 19	– Resposta à requisição contendo as perguntas a serem respondidas	29
FIGURA 20	– Diagrama que representa o arquivo XML para a comunicação com os serviços Web	30
FIGURA 21	– Código XML presente no arquivo “servicos.xml”	31
FIGURA 22	– Conversa entre o solicitante e a orquestração	33

LISTA DE TABELAS

TABELA 1	–	Agrupamentos Fonéticos do <i>Soundex</i>	25
----------	---	--	----

LISTA DE SIGLAS

UTFPR-CM	Universidade Tecnológica Federal do Paraná - Câmpus Campo Mourão
BPMN	Business Process Model and Notation
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
SOAP	Simple Object Access Protocol
HTTP	Hypertext Transfer Protocol
XML	eXtensible Markup Language
SOA	Service-Oriented Architecture
HTTPS	HyperText Transfer Protocol Secure
OMG	Object Management Group
API	Application Programming Interface

SUMÁRIO

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO	2
1.2	JUSTIFICATIVA	2
1.3	OBJETIVOS	3
1.4	OBJETIVOS ESPECÍFICOS	3
2	METODOLOGIA	5
3	REFERENCIAL TEÓRICO	7
3.1	TECNOLOGIAS UTILIZADAS	11
3.2	TRABALHOS RELACIONADOS	12
3.2.1	RobôED	13
3.2.2	Genesis System	13
3.2.3	Composição Automática de Web Services: JSHOP2 e JESS4	14
4	ANÁLISE E PROJETO DO SISTEMA	15
4.1	SERVIÇOS WEB	17
4.2	ORQUESTRADOR DE SERVIÇOS	17
4.3	SERVIÇO NOTAS POR ALUNO	18
4.4	SERVIÇO PENDÊNCIAS NA BIBLIOTECA	18
4.5	SERVIÇO CONSULTA HORÁRIO DE AULAS	19
4.6	SERVIÇO CONSULTA LABORATÓRIOS VAGOS	20
4.7	SERVIÇO RESERVA LABORATÓRIOS, SALAS DE AULA E ANFITEATRO	21
4.8	SERVIÇO REQUERIMENTO DE MATRÍCULA	22
5	DESENVOLVIMENTO DO PROJETO	23
5.1	INCORPORAÇÃO ÀS REDES SOCIAIS	23
5.2	INTERPRETAÇÃO TEXTUAL	24
5.3	ELABORAÇÃO DOS SERVIÇOS WEB	26
5.3.1	Serviços Web com Dados Fictícios	27
5.3.2	Métodos Comuns nos Serviços Web	28
5.3.3	Esquema XML	29
5.4	ORQUESTRADOR	29
5.4.1	Preparação do Orquestrador	32
5.4.2	Interpretação das solicitações	32
5.4.3	Conversa	32
6	CONCLUSÃO	34
6.1	TRABALHOS FUTUROS	34
	REFERÊNCIAS	36

1 INTRODUÇÃO

O crescente uso das tecnologias Web nos últimos anos aumentou a demanda por serviços disponibilizados pela internet. Nesse contexto, surge um novo segmento de tecnologia chamado de *Web Services*, ou em português, serviços Web. Esse conjunto de tecnologias permite a comunicação entre aplicações computacionais executando em diferentes plataformas, usando a internet e um conjunto de padrões e protocolos como meio de comunicação. Com essa nova modalidade de aplicações, cria-se um leque de possibilidades em que recursos avançados podem ser utilizados com relativa facilidade por aplicações mais modestas. A principal característica dos serviços Web é integrar pequenos serviços à execução de uma atividade maior e mais complexa, denominada como rede de serviços. Posteriormente, essa rede de serviços e sua potencialidade técnica pode ser integrada a outros sistemas ou serviços externos. Versatilidade, flexibilidade e interatividade são outras qualidades referentes aos serviços Web. Essas características são de grande benefício a um ambiente acadêmico, empresarial ou institucional.

A Universidade Tecnológica Federal do Paraná – Câmpus Campo Mourão (UTFPR-CM) figura como um ambiente de grande potencial para o uso de uma rede de serviços, sendo escolhida para o estudo da viabilidade do uso dessa tecnologia na tentativa de suprir a demanda natural de integração de serviços, atividades e eventos que compõem a relação cotidiana entre universidade, professores, acadêmicos e servidores. Reduzir a complexidade dessas relações promovendo facilidade de entendimento, agilidade e precisão, é um grande desafio para o qual o uso dos serviços Web demonstra ser uma possível solução. Essa possibilidade interessante motivou a pesquisa e a escrita deste trabalho que visa fomentar a possibilidade de construção de uma composição de serviços Web específicos para o ambiente institucional acadêmico.

O foco deste trabalho é propor uma estrutura versátil, simples, centralizada e escalável, criada com o uso de serviços Web na função de executar ações rotineiras que usuários efetuam diariamente. Criar um sistema para consumir esses serviços de forma simples e intuitiva para que os próprios usuários possam fazer uso dos seus benefícios sem maiores dificuldades de interatividade ou adaptação também é objetivo deste trabalho.

1.1 MOTIVAÇÃO

É importante conhecer e entender toda a complexidade que envolve a tecnologia dos serviços Web, transparente para programadores e usuários de sistemas de informação. Informações e serviços contidos nesse estudo podem favorecer toda a comunidade acadêmica, em virtude da simplicidade de implantação, desenvolvimento e uso. Aliada a toda sua praticidade e a possibilidade de proporcionar processamento distribuído em sistemas heterogênicos.

A metodologia para a criação de um *Software* pode ser considerada como algo complexo, afinal, demanda de um grande estudo em relação a coleta de requisitos, estudos de adaptação, estudo de tecnologias para a implementação, implantação, gerenciamento e correções de possíveis erros, e cuidados em relação às questões de segurança. Esses estudos ou etapas podem ser minimizados com o uso de serviços Web. Um serviço Web funciona como uma espécie de módulo ou segmento de *Software*. Assim, uma vez implementado, pode ser reaproveitado na construção de um novo *Software* ou utilizado por outras aplicações, independente da tecnologia usada em sua construção. A modularidade que os serviços Web proporcionam, sendo independentes de plataforma e descentralizados, pode propiciar a criação de meios de interação alternativos que vão além de formulários eletrônicos ou páginas Web. A flexibilidade proporcionada estende-se também aos dispositivos móveis ou qualquer outro meio que possa acessar a internet.

Essas características motivam a criação de aplicações que reduzam o cansaço e estresse em virtude de uso excessivo ou mau uso. Essa questão denominada usabilidade, deve ser considerada como uma importante etapa na construção de um bom *Software*, e para tal, aspectos como ergonomia e perspectivas audiovisuais devem ser estudados em detalhes. A usabilidade é definida como “capacidade que um sistema interativo oferece a seu usuário, em um determinado contexto de operação, para a realização de tarefas, de maneira eficaz, eficiente e agradável” (BASTIEN et al., 1999). Santos (2007, p. 4) ressalta ainda a influência da usabilidade na aceitação e no sucesso de um sistema computacional. Assim, este trabalho também propõe um meio alternativo de interação que utiliza *Softwares* consolidados que respeitam normas e diretrizes referentes à usabilidade.

1.2 JUSTIFICATIVA

Com a popularização da Internet surgiram novas técnicas para execução de tarefas e solução de problemas de forma globalizada, rápida e extensível. Os serviços Web são um exemplo prático dessa abordagem. Observa-se no escopo da universidade que há a dificuldade

de usabilidade no uso de sistemas acadêmicos. Esse trabalho visa propor a utilização de serviços Web para a execução de tarefas cotidianas dos acadêmicos e servidores da UTFPR-CM, buscando toda a comodidade e benefícios que essa tecnologia proporciona. É desejável o uso de aplicações já existentes, consolidadas e que seja utilizada uma sistemática que humanize¹ atividades consideradas complexas. Em um cenário considerado ideal e desejável, a interação dos usuários com o sistema deixaria de ser realizada através de formulários eletrônicos ou páginas Web e seria realizado por algo similar a uma simples conversa textual, em um mecanismo de mensagens instantâneas com um agente que simule características humanas e que seja capaz de entender e responder a uma solicitação do usuário.

1.3 OBJETIVOS

Como primeiro objetivo, este trabalho pretende criar uma composição de uma rede de serviços Web para executar tarefas específicas no ambiente institucional da UTFPR Câmpus Campo Mourão, atentando ao fato que essa rede de serviços assume um caráter simulado, com objetivo de demonstrar os benefícios que essas tecnologias podem proporcionar. Além disso, o trabalho também incorpora a composição de serviços simulados em uma ferramenta com a qual um usuário possa interagir de forma rápida e de fácil acesso.

Com um estudo mais elaborado e com uma implementação mais aprofundada, fica aberta a possibilidade de utilização do trabalho em um ambiente de produção, ou seja, com validação de dados e manipulação de informações presentes nas aplicações em uso real.

1.4 OBJETIVOS ESPECÍFICOS

Assim, tem-se por objetivos específicos:

- Propor um conjunto de serviços Web que simulem a execução de tarefas cotidianas do ambiente acadêmico;
- Elaborar um gerenciador de requisições e um orquestrador de serviços para a realização das tarefas acadêmicas utilizando os serviços Web propostos;
- Criar serviços compostos que utilizam dados fictícios com o objetivo de testar a implementação;

¹O termo humanizar presente neste trabalho refere-se ao ato de tornar a interação usuário/sistema o mais próximo possível a comunicação realizada por humanos (ato de conversar ou gesticular).

- Desenvolver e implementar um meio de comunicação entre serviços e usuários, que possa promover transparência e facilidade de acesso durante a interação dos serviços propostos;

2 METODOLOGIA

Primeiramente foram abordados os conceitos envolvidos na criação e utilização dos serviços Web. Posteriormente estudou-se as tecnologias que dão suporte à arquitetura de serviços Web como padrões de comunicação e linguagens de programação.

Fundamentado com o conhecimento obtido no estudo de tecnologias e métodos foi elaborado um conjunto de serviços Web que visam demonstrar os benefícios dessa tecnologia em processos de negócio presentes no cotidiano da UTFPR-CM.

Após a modelagem dos serviços Web simulados foi desenvolvido o orquestrador de serviços visando atender aos seguintes requisitos:

- Identificar o solicitante¹;
- Receber e analisar as requisições oriundas do solicitante;
- Interpretar qual ação o usuário deseja e verificar dentre os serviços Web registrados se há algum que atenda a solicitação;
- Fazer uma chamada ao serviço Web que teoricamente corresponde a necessidade do solicitante;
- Selecionar as questões que devem ser respondidas pelo solicitante contendo dados adicionais necessários;
- Repassar as questões individualmente ao solicitante e coletar as respostas até que todas as questões sejam respondidas;
- Invocar o serviço Web repassando as informações adicionais como parâmetro;
- Coletar a resposta gerada pelo serviço Web e repassar ao solicitante;
- Encerrar a conversação e permanece disponível para uma nova solicitação.

¹A denominação de solicitante é dada ao usuário que mantém uma conversa com o comunicador instantâneo

Em suma, podemos dizer que o orquestrador é um serviço Web cuja ação principal é interagir com os demais serviços Web publicados, em atendimento a uma determinada solicitação. Cabe ao mesmo a função de controlar todas as ações que resultarão em uma conversação.

Dessa forma foi implementado um meio de comunicação e interação com o usuário. Mediante várias possibilidades e opções quanto a uma interface de comunicação, optou-se por um comunicador instantâneo que faz a ponte de comunicação entre serviços e usuários. O uso do comunicador instantâneo tem por objetivo abstrair a complexidade do sistema, de forma que o comunicador se assemelhe a uma linha de comunicação com atitudes, percepções e ações humanas. São objetivos específicos do comunicador:

- Fazer as chamadas necessárias ao orquestrador de serviços para que o mesmo execute os serviços de acordo com o modelo estabelecido;
- Solicitar informações adicionais necessárias ao usuário quando houver a necessidade;
- Coletar as informações oriundas dos serviços e interpretá-las para formulação de uma resposta ao usuário;
- Fornecer a resposta ao usuário.

Foi necessário o uso de tecnologias e técnicas específicas para interpretar a solicitação do usuário e associá-la a um serviço existente. O uso do algoritmo fonético *Soundex* (CARR, 2010) aparece como uma alternativa interessante na busca de tal nível de abstração. Conforme a BlockWasp define: “o algoritmo *Soundex* utiliza regras que transformam sequências de caracteres em fonemas, unificando assim palavras com sons semelhantes”. Podemos desmembrar a mensagem enviada pelo usuário, separar as sequências de palavras e compará-las foneticamente com as palavras associadas aos serviços e processos registrados, gerando um nível de similaridade entre solicitação e serviços. O sistema usará a com maior nível de similaridade.

A informação será repassada ao orquestrador que se encarregará de executar uma composição dos serviços solicitados e suas dependências. A orquestração será pré-estabelecida por meio de modelo feito usando a notação BPMN.

Com o objetivo de testar o sistema, será elaborado um conjunto de testes para o orquestrador de forma a demonstrar seu funcionamento durante a composição de serviços. Também serão elaborados testes para validação do conjunto de serviços Web simulados que este trabalho propõe.

3 REFERENCIAL TEÓRICO

Nesta sessão recorreremos às pesquisas e trabalhos reconhecidamente consistentes e de excelência. Visando assim credibilidade necessária à consecução dos objetivos inicialmente propostos.

Segundo a W3C (World Wide Web Consortium), um Serviço Web é:

“...um sistema de *Software* planejado para suportar a interação máquina a máquina em uma rede. Ele tem uma interface descrita em um formato processável de máquina (especificamente WSDL). Outros sistemas interagem com um Serviço Web de maneira prescrita por seu descritor usando mensagens SOAP, tipicamente transportadas usando HTTP e uma serialização XML em conjunto com outras tecnologias relacionadas à Web.” (W3C, 2013)

A função principal do serviço Web é promover a comunicação entre duas aplicações para a realização de uma determinada tarefa independentemente da linguagem ou tecnologia com as quais foram construídas. Para que seja possível essa comunicação o serviço Web deve usar tecnologias padronizadas como, por exemplo, XML, SOAP, WSDL e HTTP, recomendações do World Wide Web Consortium (W3C).

A linguagem XML (eXtensible Markup Language) é uma linguagem de marcação genérica padronizada pelo W3C (QUIN, 2010). O XML foi criado e padronizado para ser um formato simples, baseado em texto para representar informações estruturadas. Mouta (2010, p. 85) é enfático ao afirmar que a linguagem XML tem a função específica de estruturar os dados de forma que possam ser transportados entre diferentes plataformas de maneira padronizada, diferentemente de outras linguagens ou protocolos que possuem papéis mais marcante nesse contexto.

O protocolo HTTP (Hypertext Transfer Protocol) é um protocolo de comunicação utilizado para transferência de dados em rede. De acordo com Fielding et al. (1999), o HTTP utiliza o modelo cliente/servidor, ou seja, um programa cliente estabelece uma comunicação com um outro programa servidor e envia uma requisição contendo a URI, informações do cliente e o

conteúdo do chamado corpo da mensagem. O servidor responde com uma linha de status incluindo sua versão de protocolo e um código de operação bem sucedida ou um código de erro, seguido pelas informações do servidor, meta informações da entidade e possível conteúdo no corpo da mensagem. Após o envio da resposta pelo servidor, encerra-se a conexão estabelecida.

Para que seja possível a execução de rotinas ou procedimentos é necessário padronizar as informações de interação entre serviços e aplicações. Tal função fica a cargo do SOAP – Protocolo Simples de Acesso a Objetos, do inglês *Simple Object Access Protocol* – que é um protocolo projetado com o objetivo específico de invocar aplicações remotamente através de chamadas remotas de procedimentos ou troca de mensagens em um ambiente independente de plataforma e linguagem de programação (ERL, 2005).

Finalmente, para descrever a interface de serviços remotos de maneira estruturada usamos o WSDL (Web Services Description Language). Christensen et al. (2001) define o WSDL como um documento, escrito em XML, que descreve a localização dos serviços Web e as operações ou métodos que compõem o mesmo. Enquanto o SOAP especifica a comunicação entre o cliente e o servidor, o WSDL descreve os serviços oferecidos.

De maneira genérica, podemos afirmar que a comunicação entre aplicações é composta por um processo de envio e recepção de mensagens estruturadas por XML. As chamadas à uma determinada operação são codificadas no protocolo SOAP e os serviços são descritos usando a linguagem WSDL. Por fim, o meio de transporte fica a cargo do protocolo HTTP.

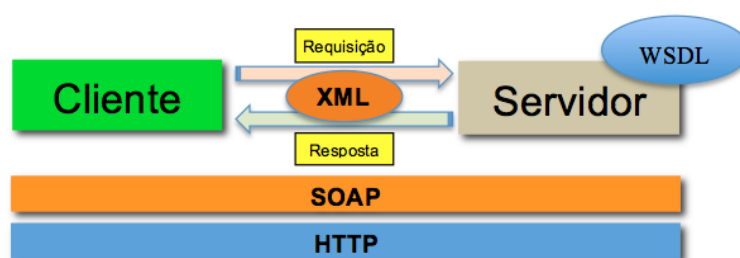


Figura 1: Estrutura de comunicação de um serviço Web

Fonte: Autoria própria

A Figura 1 ilustra as estruturas que compõem um Web service. Uma das principais características dos serviços Web é sua simplicidade. Geralmente um serviço resume-se a um método ou função. É necessária a presença de dois contextos: o Cliente (também denominado de Consumidor) e o Servidor (também denominado de Provedor). Clientes efetuam chamadas aos serviços disponíveis nos servidores usando solicitações em XML. As solicitações são processadas e utilizam uma bases de dados para geração de respostas, também em XML, interpretadas pelos clientes. Ambos (cliente e servidor) utilizam como forma de comunicação padrões e tecnologias simples, presentes em diversas linguagens de programação e sistemas

operacionais. A simplicidade se resume à comunicação entre os contextos, pois a geração e o uso das informações são de responsabilidade dos clientes e servidores e dependentes de suas aplicações e bases de dados. Há ainda a presença de outros protocolos, como o SOAP, responsável por garantir a troca de informações entre sistemas e o WSDL que descreve os tipos de dados trocados nas mensagens e efetua o mapeamento entre as mensagens e seus protocolos de transporte. A integração desses protocolos cria uma independência quanto aos protocolos de transporte, permitindo o uso do protocolo de transporte HTTP.

Nota-se que todas as ações relacionadas aos serviços Web são síncronas, mesmo porque os próprios padrões utilizados em sua criação (SOAP, WSDL e até mesmo o HTTP) possuem essa característica, são restritos a requisições e eventuais respostas. Deve-se ressaltar que o conceito de sincronia no protocolo HTTP está diretamente ligado ao conceito de temporalidade. Embora trabalhe de maneira síncrona (toda requisição obtém resposta específica) entre uma requisição e sua resposta podem ocorrer outros eventos (requisições e respostas do mesmo serviço), o que torna um serviço Web síncrono e atemporal. Esta é uma característica que limita o potencial dos serviços Web a ações simples, de requisição e resposta, tornando-os ineficientes por si só na realização de processos mais complexos.

O conceito de composição de serviços está relacionado a um conjunto de serviços Web independentes sendo utilizados de maneira conjunta. Para gerar uma resposta um serviço pode, inclusive, fazer uso de outros serviços Web. Por exemplo, para o cálculo de frete para uma compra, pode ser necessário obter informações a respeito do endereço informado utilizando um serviço Web específico (Correios, por exemplo). Temos assim um novo conceito de composição de serviços Web. Para que essa relação ocorra de maneira adequada é essencial estabelecer na composição, quais serviços participam e qual sua dependência, o que determina uma ordem de execução de serviços. No caso de dependência entre os serviços, a composição deve definir qual o papel de cada serviço durante a execução de um determinado processo. Nesse caso, a composição de serviços necessita diretamente de uma ferramenta que atue no papel de coordenação dos serviços, garantindo sua correta execução. Dois métodos de composição de serviços se destacam atualmente: a orquestração e a coreografia.

Coreografia e orquestração são dois conceitos distintos. Peltz (2003) define orquestração como um processo de negócio executável que pode interagir com serviços Web internos e/ou externos. As interações ocorrem em nível de mensagens, e incluem a lógica de negócio e a ordem de execução das tarefas, com controle do ponto de vista de uma das partes, agindo como um árbitro. Victoriano e Cardoso (2010) definem coreografia como um modelo de interação em que cada participante envolvidos no processo de negócio é responsável pela execução de suas próprias tarefas não havendo a presença de um controlador central. Cada participante da com-

posição sabe exatamente quando e como atuar durante a execução, assim os serviços devem ser nitidamente mais elaborados para poderem atuar de forma correta durante a execução. O conceito de orquestração difere do conceito de coreografia justamente pela existência de uma figura centralizadora que dita às regras durante a execução (como um maestro de uma orquestra, logo a origem do nome “Orquestração”) ligando diversos serviços uns aos outros para a realização de uma determinada função. Como ilustrado nas Figuras 2 e 3.

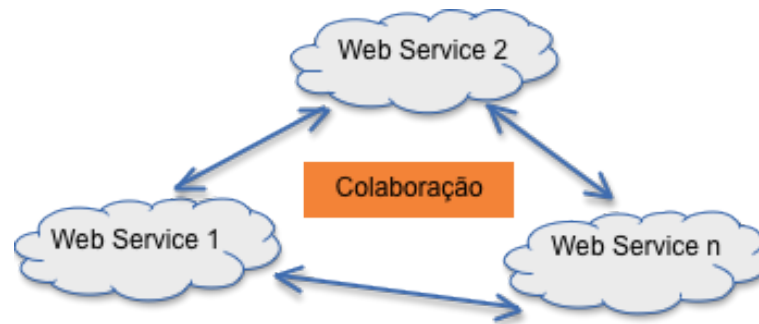


Figura 2: Coreografia de serviços Web

Fonte: Adaptado de Victoriano e Cardoso (2010)

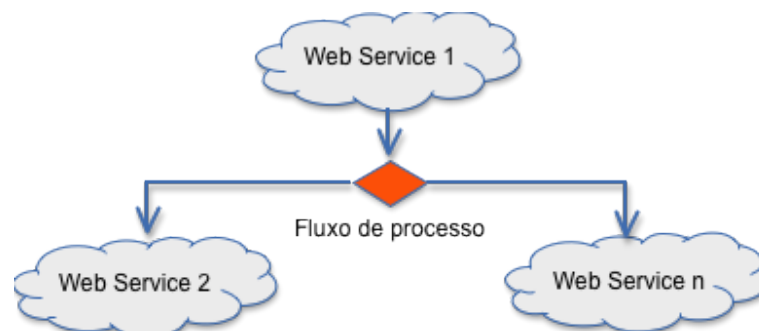


Figura 3: Orquestração de serviços Web

Fonte: Adaptado de Victoriano e Cardoso (2010)

Inicialmente é preciso entender sobre os pormenores dos serviços Web, suas teorias e padrões. Damasceno (2009) define um serviço como uma entidade que provê alguma funcionalidade. A partir desses serviços e suas interfaces, novos serviços podem ser construídos. Quando um serviço é associado a ambientes computacionais surge a ideia da computação orientada a serviços, ou seja, funcionalidades implementadas por *Softwares* disponibilizados na forma de serviços. Logo surge o Arquitetura Orientada a Serviços (SOA - Service-Oriented Architecture). Papazoglou et al. (2007, p. 38) refere-se a SOA como um paradigma para criação de componentes de aplicações em uma rede de serviços flexíveis que possibilitem a elaboração de processos de negócio dinâmicos e ágeis, abrangendo plataformas computacionais. Essa abordagem é independente de linguagens de programação específicas ou sistemas operacionais. Um exemplo dessa abordagem são os serviços Web.

Os serviços Web constituem a tecnologia mais apropriada para desenvolvimento de aplicações baseadas em SOA. Por usarem protocolos específicos para realizar a comunicação (XML, SOAP, WSDL, HTTP) são totalmente independentes de sistemas operacionais e linguagens de programação. Além disso, mantém um baixo acoplamento com outros serviços e aplicações. Serviços Web são uma das soluções utilizadas na integração e comunicação entre sistemas diferentes, permitindo a sua dinamização, o que garante maior segurança, uma vez que, dispensa a intervenção humana no processo de troca de informações. Os serviços Web obedecem a um formato XML e são encapsulados pelo protocolo SOAP. O transporte dos dados fica a cargo do protocolo HTTP ou, HTTPS no caso de conexões seguras. Para que haja a validação das chamadas dos métodos é necessário descrever os serviços usando a especificação WSDL que assim como o SOAP segue um formato padrão escrito em XML.

A arquitetura de um serviço Web é caracterizada pela sua simplicidade, podendo ser resumido à chamada de um método. Essa característica seria um obstáculo ao uso dos serviços Web em complexos processos de negócio que exigem dezenas de métodos e informações para a sua concepção (forte acoplamento e interdependência). Damasceno (2009) define composição de serviços como um sistema complexo que pode ser desenvolvido utilizando serviços mais básicos. Diversos serviços Web podem ser combinados para a composição de um novo serviço.

Pensando em composição de serviços, a modelagem é uma parte muito importante na criação dos processos de negócio por transcrever em uma representação gráfica toda a complexidade e semântica na realização de uma determinada tarefa. A BPMN (*Business Process Model and Notation* ou do português Notação de Modelagem de Processos de Negócio) (OMG, 2011) é uma linguagem de modelagem muito similar à UML que fornece uma anotação gráfica das diferentes etapas para confecção de processos de negócio. O OMG (*Object Management Group*) é atualmente o responsável por manter e padronizar a BPMN. Em termos gerais podemos conceituar a BPMN como uma notação utilizada para modelar processos de negócio, ou seja, é o conjunto de símbolos que podem ser usados para descrever um processo. A BPMN foi utilizada para modelar os processos de negócio propostos, bem como toda a estrutura de dependências entre os mesmos. Um exemplo de um processo modelado pela BPMN pode ser observado na Figura 4:

3.1 TECNOLOGIAS UTILIZADAS

Para a implementação do orquestrador, dos outros serviços Web e do servidor de mensagens instantâneas foi escolhida a linguagem Java, apoiado por APIs e arcabouços específicos para persistência em banco de dados e módulos de expansão específicos para a criação e publi-

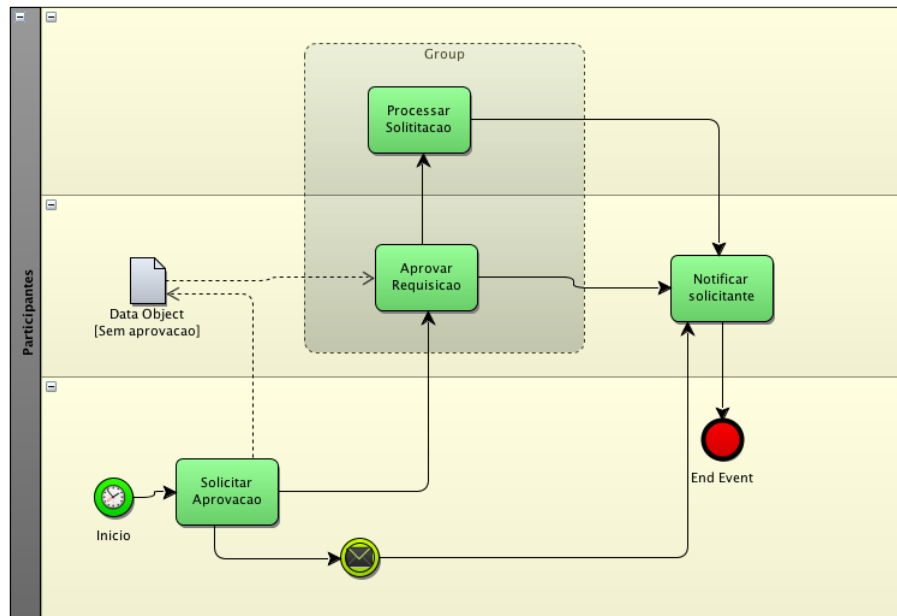


Figura 4: Exemplo de processo usando BPMN

Fonte: Autoria própria

cação dos serviços Web. A escolha da linguagem de programação Java foi pela facilidade de integração com diversos arcabouços que agilizam tarefas rotineiras.

O sistema proposto é formado por serviços Web publicados dentro de um servidor de aplicações Java denominado *Apache Tomcat* desenvolvido pela *Apache Software Foundation* (TOMCAT, 2013).

O interpretador ou orquestrador de solicitações será apoiado pelo algoritmo fonético *Soundex* (CARR, 2010) para que seja capaz de distinguir qual serviço invocar, interpretando a solicitação enviada pelo cliente.

O comunicador instantâneo foi inicialmente baseado na API Java MSN API – desenvolvida por Leov (2008) e visa formar um elo de comunicação entre cliente e o sistema utilizando comunicadores instantâneos. Ao longo do desenvolvimento do trabalho o MSN foi descontinuado e passamos a adotar o comunicador instantânea da rede social *Facebook Chat* por meio da API denominada *Smack* (COLLIER, 2013).

3.2 TRABALHOS RELACIONADOS

Esta seção descreve alguns trabalhos relacionados que ofereceram alguma contribuição para o desenvolvimento desse trabalho. Maiores informações podem ser obtidas nas referências indicadas.

3.2.1 ROBÔED

O RobôED é um projeto criado pela inSite¹ para o Ministério de Minas e Energia, é uma representação muito similar ao sistema proposto para consumir as informações geradas pela composição de serviços, justamente por oferecer a interatividade e a similaridade com percepções humanas, características desejadas para a aplicação final. A Figura: 5 mostra a forma como o RobôED apresenta-se.



Figura 5: Tela do sistema RobôED
Fonte: inSite

Trata-se de um projeto interessante. Não apenas por sua idéia original de promover ações que visam promover a sustentabilidade e preservação ambiental, mas, também pelo fato de alcançar um nível de precisão quanto a interpretação de frases bem elevado. Segundo a inSite (2013) o RobôED é capaz de reconhecer e responder mais de 10.000 tipos de frases. Além, de alcançar um índice de 95% de acertos nas respostas.

3.2.2 GENESIS SYSTEM

Neste trabalho Brandão (2012), exalta questões como componentes de serviços que podem ser orquestrados em tempo de execução que venham a atender necessidades de processos de negócios específicos, complexos e dinâmicos. Esse trabalho foi de grande utilidade na conceptualização de questões como composição, promoção, escalonamento e controle de serviços. O autor ainda propõe formas de orquestrar serviços e suas estruturas básicas de sistemas.

¹A inSite é uma empresa de desenvolvimento de *Software* fundada em 1995 e destaca-se por criar soluções voltadas a comunicação entre empresas e clientes como *chats*, *Webforums*, *Webmail*, *chatterbots* dentre outras.

como *Domain System (DS)*, *Public Subscribe System (PSS)*, *Distributed Hash Table System (DHTS)*, *Generic Indirection Resolution System (GIRS)* e *Search and Discovery System (SDS)*.

3.2.3 COMPOSIÇÃO AUTOMÁTICA DE WEB SERVICES: JSHOP2 E JESS4

Em (SILVA, 2007) são abordadas questões como o tratamento, planejamento e uso de serviços envolvidos na realização de um determinado processo, estudando mecanismos que venham a automatizar a composição de serviços Web. Silva (2007) resalta concepções básicas necessárias para a concepção de serviços e vai mais além abordando técnicas de planejamento de composições com o uso de ferramentas planejadoras como o JSHOP2 e JESS4. Esse trabalho contribuiu e muito para o aprendizado das técnicas de composição de serviços assim como abre precedentes quanto a possibilidade de um projeto futuro fundamentado a uso de planejadores de serviços.

4 ANÁLISE E PROJETO DO SISTEMA

Com a intenção de demonstrar de forma prática a potencialidade da arquitetura SOA pode oferecer ao ambiente da universidade, esse estudo propõe a criação de uma composição de serviços de forma simulada e em caráter experimental, com a intenção de analisá-los e indicá-los a uma eventual implantação em nível de produção no futuro. Essa composição levará em consideração os conceitos de orquestração uma vez que uma ferramenta principal assume o papel de coordenador de execução das requisições de serviços.

Alguns métodos para a realização da comunicação usuário/processo foram considerados e optou-se pela utilização de um cliente de mensagens instantâneas, que fica com o papel de interface de comunicação entre usuários e respectivos serviços.

Os usuários conectam-se utilizando o *Facebook Chat* e o coordenador de requisições apresenta-se como mais um usuário disponível à espera de uma solicitação de serviço. Para tal, o coordenador de requisições reconhece alguns comandos específicos que disparam chamadas ao orquestrador de serviços. Por sua vez, esse processa as solicitações e provê uma resposta usando os serviços Web necessários dentre os disponíveis.

O cenário de uso proposto (Figura 6) seria: um usuário (aluno, professor, ou funcionário) em um comunicador instantâneo adicionaria o endereço de e-mail do sistema a seus contatos. Como o sistema tem um comportamento idêntico ao de um usuário comum de comunicador instantâneo, esse passa a estar disponível para troca de diálogos assim que adicionado aos contatos. Quando o usuário necessitar de alguma informação ou serviço ele inicia uma bate-papo com o sistema e digita o serviço desejado.

Uma vez conectado com a ferramenta, outro cenário de uso poderia ser: um usuário digita: “desejo ver minhas notas”; o sistema deve interpretar a solicitação e verificar se dentre os serviços disponíveis há algum que se enquadre à solicitação. Uma vez identificado o serviço solicitado, o comunicador repassa ao orquestrador essa informação, que por sua vez, invoca os serviços Web necessários para a realização do processo. Durante a execução podem existir interações com o cliente caso algum processo exija alguma informação extra para gerar uma

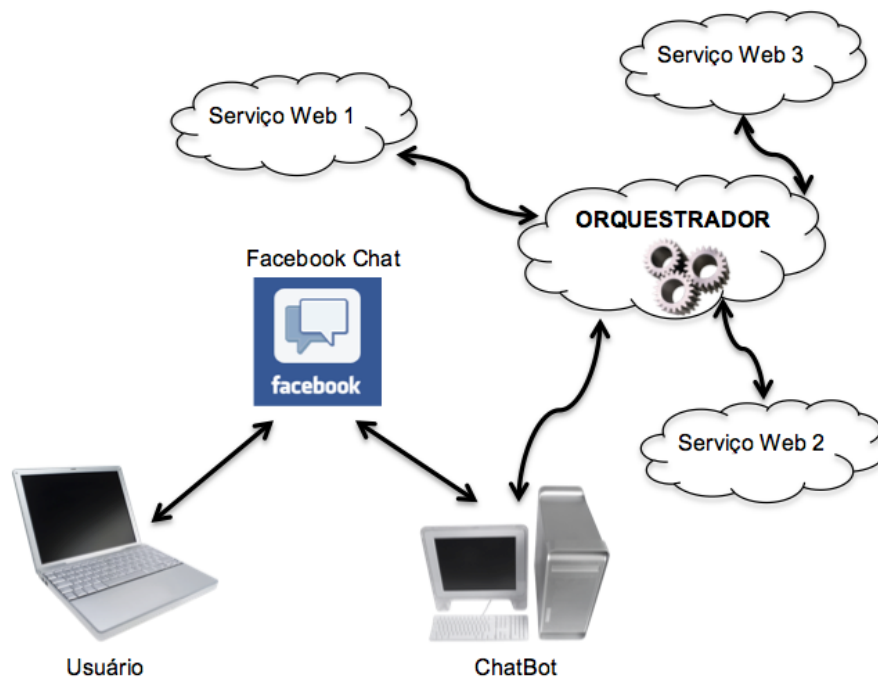


Figura 6: Diagrama ilustrando o sistema proposto
Fonte: Autoria própria

resposta. Quando pronta, essa resposta será encaminhada pelo orquestrador ao comunicador instantâneo que repassará a resposta ao usuário.

Para o desenvolvimento da ferramenta, alguns processos presentes atualmente na universidade serão recriados de forma simulada. Dentre eles o processo de matrícula, consulta de eventuais pendências no acervo bibliográfico, consulta ao horário de aulas e ensalamento, consulta salas ou laboratórios vagos, listagem de contatos e informações sobre alunos e grupos de estudo, dentre outros. Para a realização dessas tarefas é necessária a utilização de um conjunto de serviços Web integrados. Por exemplo, como pré-requisito para que um aluno possa se matricular é necessário que o mesmo não possua pendências na biblioteca, existindo assim a necessidade do uso de um outro serviço, o de consulta à biblioteca. O mesmo aluno ao final da matrícula deve ser informado sobre seu horário de aulas e ensalamento.

Todos os processos devem estar à disposição de qualquer membro da instituição, porém a segurança quanto à integridade dos dados pessoais deve ser implementada. Dessa forma, além de manter o usuário seguro, é necessário o uso de tecnologias fáceis em que todos possam acessar os serviços disponíveis de forma dinâmica e intuitiva, deixando transparente ao usuário final a complexidade da realização dessas tarefas.

As possibilidades de uso para esta arquitetura são expansíveis, existindo ainda a possibilidade futura de conjuntos de serviços Web mais elaborados e complexos como, por exemplo,

um sistema para gestão de conteúdo didático e um gestor de frequências de alunos.

4.1 SERVIÇOS WEB

O objetivo principal desse trabalho é propor um conjunto de serviços Web para uso institucional. Cada serviço Web é considerado independente quando a sua efetiva execução não depende de informações oriundas de outros serviços. Em contrapartida, um serviço Web é considerado dependente quando depende de informações externas oriundas de outros serviços. A ideia de orquestração de serviços parte do princípio de que há um agente centralizado cuja principal função é gerenciar e selecionar os serviços Web necessários para a execução de uma determinada tarefa. Em nosso caso, esse serviço Web é denominado orquestrador. O papel desse orquestrador é receber solicitações por parte dos clientes, interpretar e capturar informações vindas dos serviços Web, montar uma resposta baseada nas informações recebidas e enviar a resposta ao solicitante.

O orquestrador será conhecido como robô (pois assume o papel de um ser humano na interface do comunicador instantâneo), e recebe o papel de coordenador de requisições para a realização de um determinado processo. Considerando a possibilidade de interdependência de serviços é necessário que este coordenador de requisições seja capaz de solicitar novas informações junto aos usuários, quando necessário, para obter respostas para serviços subsequentes. Alguns serviços Web foram projetados de forma experimental com intuito de testar o funcionamento desta composição de serviços. Uma vez que os serviços são experimentais e com o objetivo de simular atividades do cotidiano, os dados serão previamente preparados para demonstrar possíveis situações de uso do sistema. Uma breve descrição dos serviços é feita a seguir.

4.2 ORQUESTRADOR DE SERVIÇOS

O paradigma escolhido para a concepção desse trabalho é voltado à orquestração de serviços. Como parte integrante dessa operação foi desenvolvido um serviço Web Java cuja a função principal é gerenciar todo o conjunto de serviços Web no atendimento a uma determinada requisição.

O diagrama da Figura 7 descreve as classes que compõem o orquestrador. Podemos destacar a dois métodos principais na classe Orquestrador: *Conversa* (que é a ponte de comunicação entre os serviços e o solicitante) e o *SearchService* (que seleciona os serviços necessários para a realização de uma determinada tarefa dentre os serviços que compõem a orquestração,

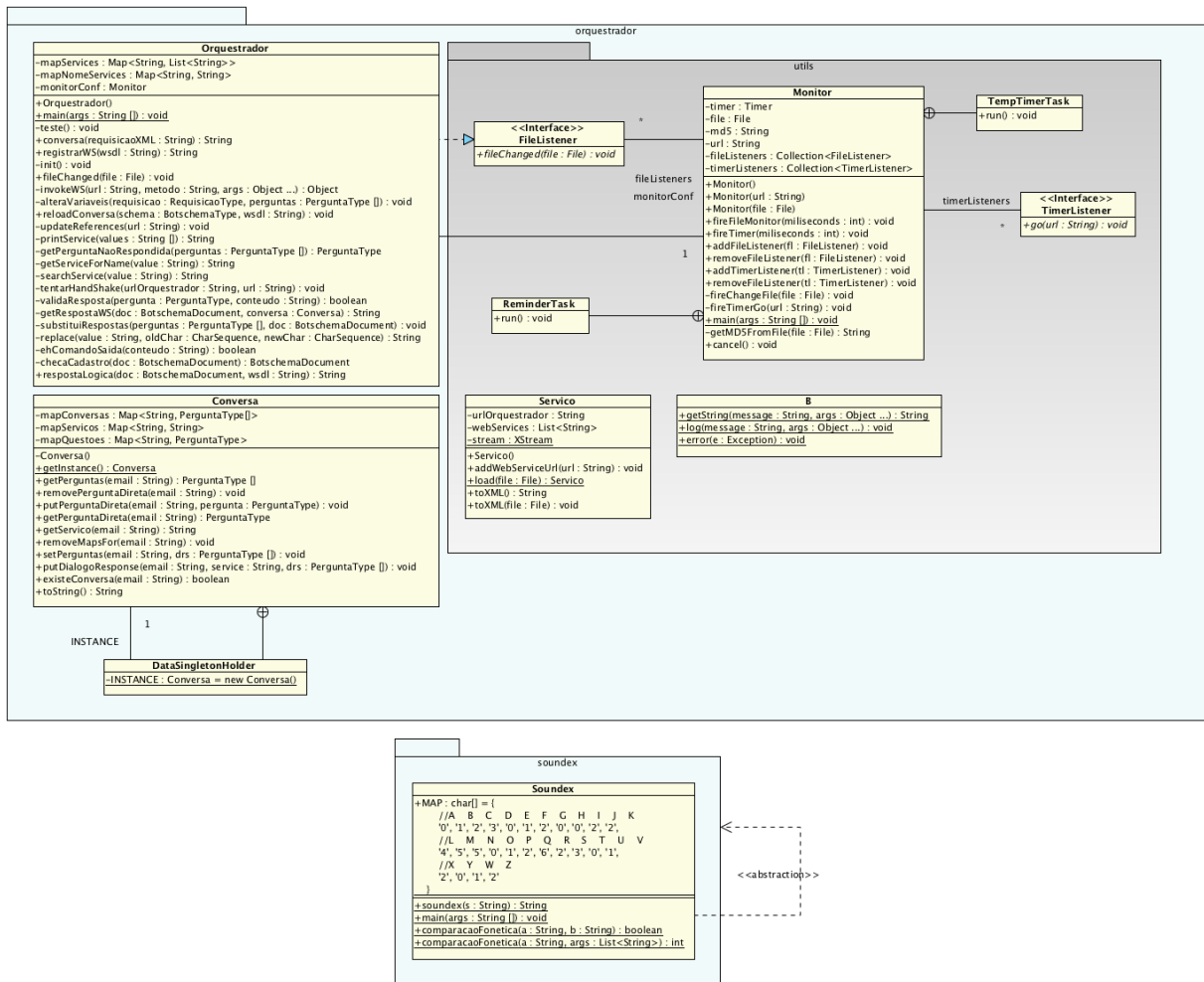


Figura 7: Diagrama de classes do orquestrador proposto

Fonte: Autoria própria

baseando-se no algoritmo fonético *Soundex*).

4.3 SERVIÇO NOTAS POR ALUNO

O Serviço Web “Notas por Aluno” (Figura 8) possibilita que o aluno acompanhe os resultados de suas avaliações. Hoje este acompanhamento é realizado de forma desarmonizada e a critério de cada professor. Esse serviço foi pensado com intuito de padronizar a divulgação de resultados de avaliações, de forma prática rápida e individualizada garantindo a privacidade dos dados.

4.4 SERVIÇO PENDÊNCIAS NA BIBLIOTECA

O serviço Web “Pendência na Biblioteca” (Figura 9) tem por objetivo auxiliar usuários na obtenção de informações quanto a pendências junto à biblioteca da instituição. Essa ferra-

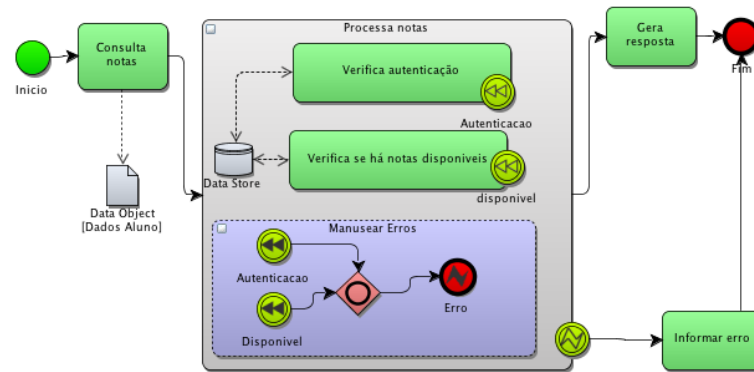


Figura 8: Diagrama BPMN do serviço “Notas por Aluno”
Fonte: Autoria própria

menta pode ser de grande valor pois muitas ações dentro da universidade exigem que o aluno não tenha pendências. O processo de matrícula é um exemplo prático dessa regra. Um aluno não pode efetuar requerimento de matrícula caso tenha pendências junto ao acervo bibliográfico da instituição. Outro fator a ser analisado é que o aluno está propenso ao pagamento de multas em caso de atraso na devolução de livros. Em decorrência disso, esse serviço pode servir como ferramenta para acompanhamento de prazos de entrega evitando assim possíveis atrasos e eventualmente o pagamento de multas.

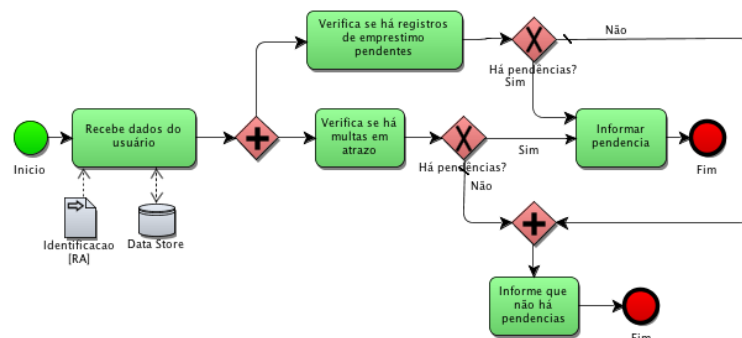


Figura 9: Diagrama BPMN do serviço “Pendências na Biblioteca”
Fonte: Autoria própria

4.5 SERVIÇO CONSULTA HORÁRIO DE AULAS

Todo início de semestre existe uma procura pelos horários das aulas, bem como salas de aula ou laboratórios, levando os alunos a procurarem essas informações junto à secretaria do câmpus. Essa ação pode sobrecarregar o setor responsável. Assim, o serviço Web “Consulta Horário de Aulas” (Figura 10) visa atender de forma rápida e prática as solicitações dos alunos. Todos poderiam ser informados sobre alterações quanto ao ensalamento ou troca de aulas, evitando assim informações desencontradas.

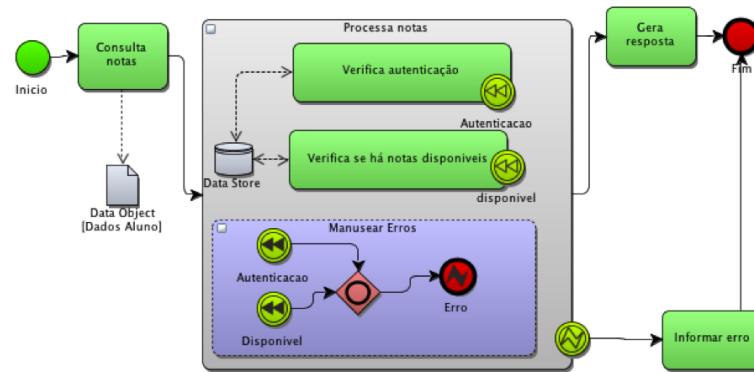


Figura 10: Diagrama BPMN do serviço “Consulta Horário de Aulas”
Fonte: Autoria própria

4.6 SERVIÇO CONSULTA LABORATÓRIOS VAGOS

O serviço Web “Consulta Laboratórios Vagos” (Figura 11) visa atender aos casos em que os alunos necessitam fazer uso dos laboratórios, porém nem sempre há laboratórios vagos. A criação de um serviço que procure por laboratórios vagos é extremamente interessante, pois um aluno que necessitar usar computadores pode fazer uso desse. Um professor que desejar reservar um laboratório pode (com o uso da ferramenta) consultar qual laboratório estará vago para uma determinada data. Funcionários responsáveis pela manutenção dos computadores podem consultar a vacância dos laboratórios para realizarem serviços específicos como instalações ou atualizações. Assim como os responsáveis pela limpeza podem realizar seus serviços em um momento em que não haverá aulas.

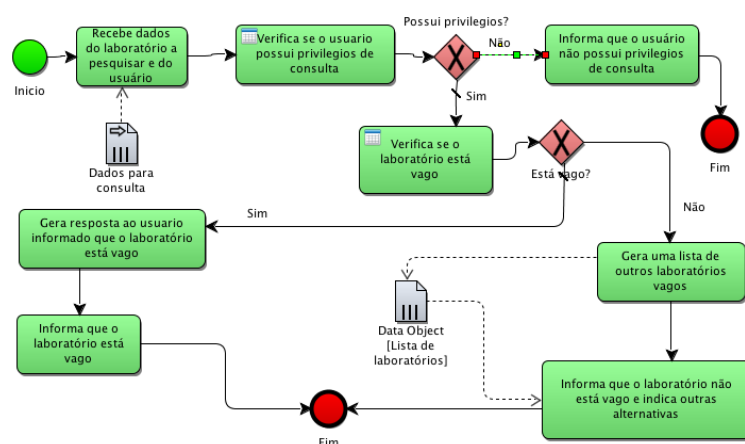


Figura 11: Diagrama BPMN do serviço “Laboratórios Vagos”
Fonte: Autoria própria

4.7 SERVIÇO RESERVA LABORATÓRIOS, SALAS DE AULA E ANFITEATRO

Para a reserva de laboratórios é necessário consultar se o ambiente está vago para aquela determinada hora. Assim, o serviço web “Reserva Laboratórios, Salas de Aula e Anfiteatro” (Figura 12) faz uso do serviço Web “Consultar Laboratório Vago”. Nesse caso há uma relação de dependência entre os serviços, gerenciada pelo orquestrador.

Toda a complexidade por trás dessa tarefa é transparente ao usuário final. Há apenas uma requisição e uma resposta que traz consigo um protocolo de serviço informando sucesso ou erro pela operação realizada. Abstraindo todo o processo que em um primeiro momento verifica se há laboratórios vagos, indica o laboratório para uso, grava o evento no banco de dados marcando o mesmo como indisponível para outras eventuais consultas sobre reservas.

Um serviço similar ao Reservar Laboratórios, é o serviço de reserva do anfiteatro, no qual professores podem solicitar uma reserva para uso. Esse serviço, porém denota maior complexidade a mais, pois se houver a necessidade de uso de multimídia ou apoio técnico, há uma demanda por auxílio específico por parte do técnico administrativo responsável. Para tanto, é necessário que este serviço não apenas efetue a reserva, mas também solicite servidores para auxílio de ordem técnica. O serviço também deve evitar o conflito de datas entre eventos, ou seja, não podem ocorrer reservas do anfiteatro para mais de um professor na mesma data e horário.

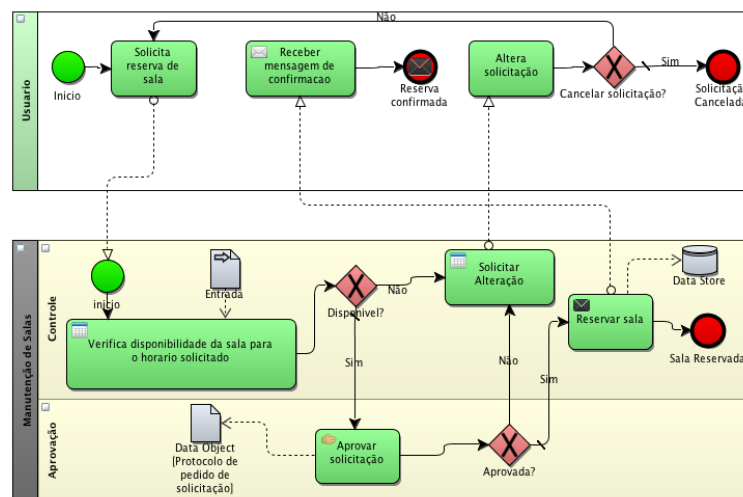


Figura 12: Diagrama BPMN do serviço “Reserva de Infraestrutura”

Fonte: Autoria própria

4.8 SERVIÇO REQUERIMENTO DE MATRÍCULA

O processo de matrícula demonstra ser um dos principais realizados pelos alunos em sua jornada pela instituição. Esse processo é complexo, seja pelos passos realizados pelo acadêmico quanto pela infraestrutura de TI que sofre com a intensa quantidade de requisições simultâneas. A complexidade envolvida durante a realização da matrícula pode gerar atrasos e frustrações ao acadêmico. O serviço Web “Requerimento de Matrícula” (Figura 13) visa facilitar esta tarefa, dando mais mobilidade aos alunos. Como dito anteriormente o processo de requerimento exige que o aluno não possua pendências junto à biblioteca do câmpus, assim ele estabelece uma dependência de informações provenientes do serviço Web “Pendências Biblioteca”.

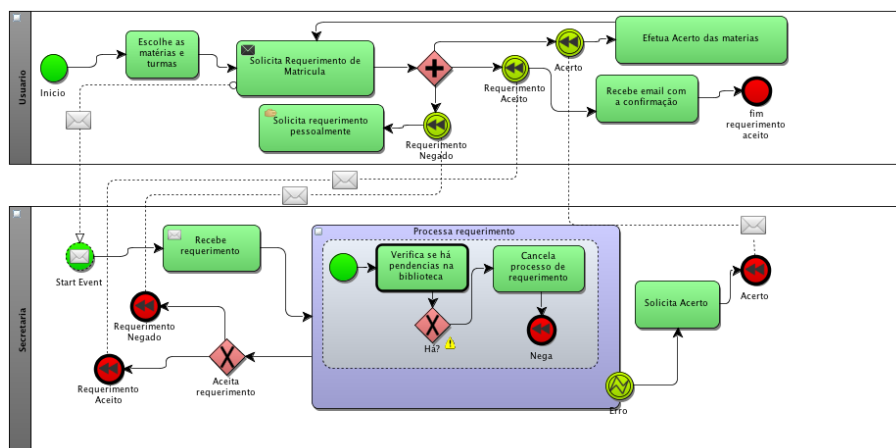


Figura 13: Diagrama BPMN do serviço “Requerimento de Matrícula”

Fonte: Autoria própria

Esse serviço não visa a substituição da sistemática de matrícula atual, mas sim, para servir como uma ferramenta de auxílio ao acadêmico. Além de amenizar o fluxo excessivo de informações da infraestrutura responsável pela realização dessa tarefa, toda a parte de validação, seleção de turmas e disciplinas, ficaria a cargo de um conjunto de serviços Web. O acadêmico receberia como resposta todas as informações para a realização efetiva de sua matrícula e o sistema de matrícula da UTFPR receberia os dados da matrícula livre de inconsistências com a expectativa de tornar o processo mais ágil. Dessa forma, existe também a expectativa de redução na complexidade da operação de matrícula do acadêmico.

5 DESENVOLVIMENTO DO PROJETO

Neste capítulo são serão abordados os detalhes da etapa do desenvolvimento do sistema proposto bem como detalhes sobre as tecnologias utilizadas e dificuldades encontradas. O objetivo principal desse trabalho é propor uma composição de serviços Web e demonstrar através de simulações como o uso desses serviços pode ser útil ao ambiente institucional. A principal dificuldade técnica foi a exigência de conhecimento específico sobre as funcionalidades ou resultados dos serviços. Apesar de relativamente simples, essa tarefa não está ao alcance de todos no âmbito da universidade. Sendo assim, pensou-se em uma maneira de levar essas funcionalidades aos mais diversos níveis de usuários. O uso de composição de serviços Web agregado às redes sociais despontou como uma alternativa interessante.

5.1 INCORPORAÇÃO ÀS REDES SOCIAIS

Tomaél et al. (2005) enfatiza a importância das redes sociais ao compartilhamento de informações e do conhecimento mediante aos atores que as integram, e ainda destaca o fato de que natureza humana interliga pessoas e estruturas a uma sociedade em rede.

A junção entre serviços Web, redes sociais e comunicadores instantâneos pode ser considerado o elo para a melhoria de usabilidade entre serviços e usuários estabelecendo um meio de conversação fácil e intuitivo entre ambos. E em pesquisa de alternativas tecnológicas encontramos uma API chamada RoboMSN desenvolvida por Leov (2008) que possibilitava a integração de aplicativos desenvolvidos em Java à rede social MSN ou *Windows Live Messenger* mantida pela *Microsoft*. Inicialmente a aplicação foi construída com sucesso e atendia prontamente aos objetivos propostos. Entretanto, durante o desenvolvimento, a plataforma MSN foi descontinuada pela *Microsoft* e em seguida incorporada e substituída em definitivo pelo *Skype* em 27 de maio de 2013. Esses eventos exigiram que o protótipo desenvolvido fosse adaptado à uma outra plataforma de comunicação. Mesmo sendo possível a adaptação para a plataforma *Skype*, nossa opção foi pelo comunicador instantâneo do Facebook, o *Facebook Chat* justamente por ter a maior rede social da atualidade e possuir grande abrangência no Brasil.

Assim com o uso da *Smack API* desenvolvida por Collier (2013) houve a possibilidade de usar o *Facebook Chat* como meio de comunicação entre os serviços e os usuários. Mesmo mostrando-se mais limitada tecnologicamente que a API para MSN, a *Smack API* atendeu aos requisitos, apesar de algumas funcionalidades antes automatizadas pela API anterior exigirem alterações manuais como, por exemplo, a inclusão de novos usuários, a aceitação das solicitações de amizade ou a alteração de perfil. Dessa forma foi construída uma aplicação que estabelece uma comunicação direta entre o orquestrador e serviço do *Facebook Chat*, abstraindo completamente do usuário final qualquer ação computacional necessária para usar os serviços Web propostos. A Figura 14 ilustra a forma como o sistema se apresenta ao usuário final.

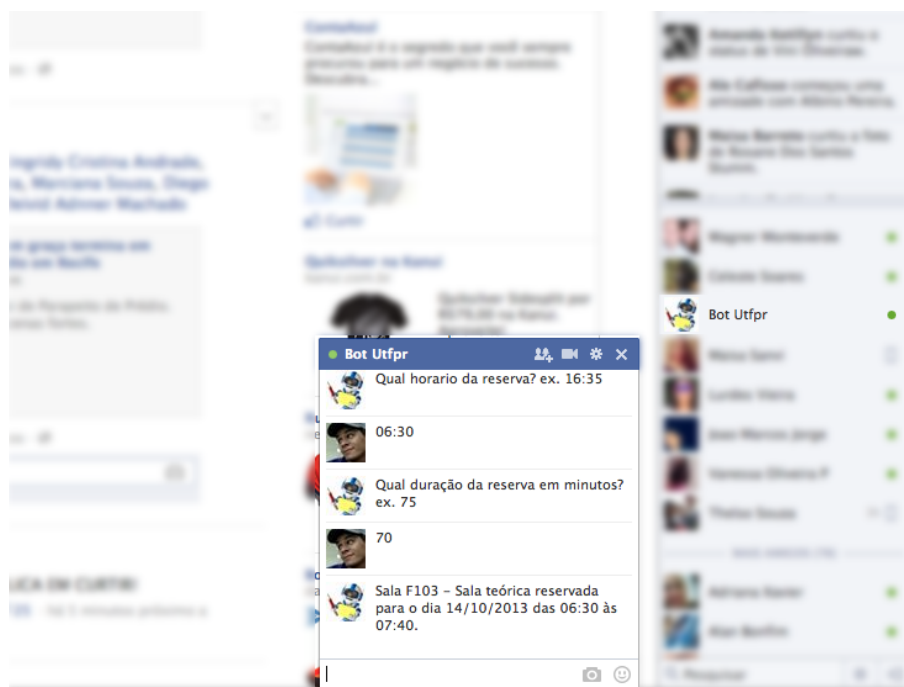


Figura 14: Exemplo de comunicação entre usuário e o sistema proposto
Fonte: Facebook

5.2 INTERPRETAÇÃO TEXTUAL

Após definir um meio de comunicação que atendesse aos requisitos preestabelecidos surgiu um novo problema em relação à interpretação das solicitações. Foi preciso considerar a forma grafo linguística adotada por chats, blogs e redes sociais e tentar processá-la e associá-la a algum serviço ou conjunto de serviços na tentativa de estabelecer uma comunicação. Komesu (2007) aborda claramente esse tema definindo como “internetês” ou o português falado na internet. Tal questão amplamente discutida e até mesmo polemizada por defensores da boa escrita e linguagem formal não faz parte do estudo abordado nesse trabalho porém, caracteriza-se como um difícil obstáculo a ser contornado tendo em vista que sistemas computacionais prezam pela

2	C, S, K, G, J, Q, X, Z
3	D, T
4	L
5	M, N
6	R

Tabela 1: Agrupamentos Fonéticos do Soundex

Fonte: Autoria própria.

precisão e lógica.

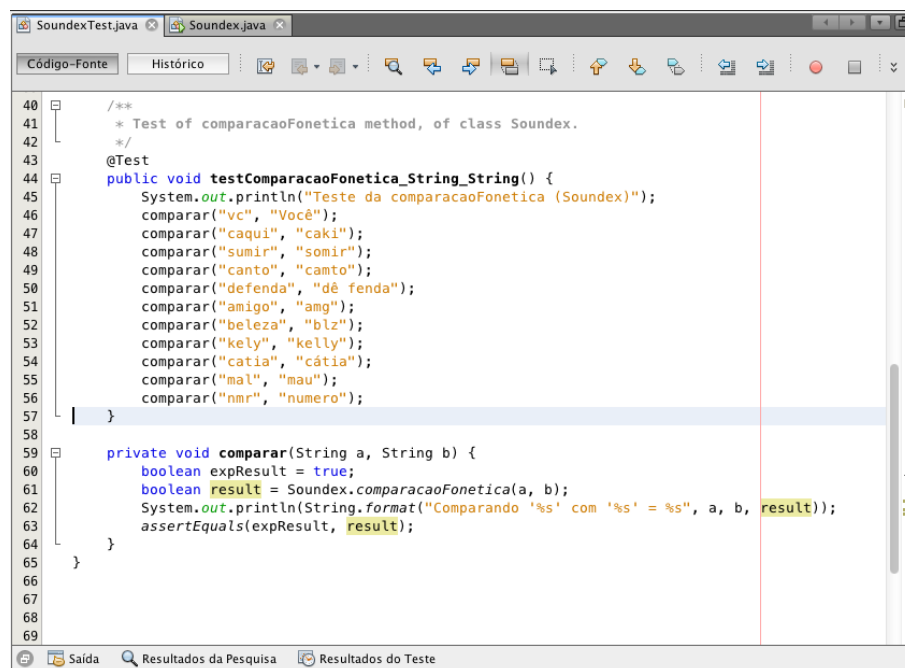
A solução encontrada ou ao menos uma tentativa de amenizar o problema foi a utilização de um conjunto de palavras-chave associadas diretamente a cada serviço. Essas palavras identifica os serviços pela chamadas diretas a cada serviço. Por exemplo: o serviço “Consultar Notas” teria como palavras-chave (consultar, ver, notas, minhas) assim um usuário que desejasse ser atendido pelo serviço poderia solicitar algo digitando “ver minhas notas” ou “desejo consultar minhas notas” e o comparativo entre a solicitação e as palavras-chave resultam em um índice de similaridade que identifica o serviço que atenda a solicitação. É no momento da comparação que surge o problema discutido nessa subseção. Não é previsível que um usuário digite uma solicitação com a precisão citada acima; o mesmo pode usar abreviações, podem ocorrer erros de português ou simplesmente fazer uso do “internetês”, impossibilitando assim o simples uso de uma comparação lógica entre as palavras, interferindo diretamente na precisão da interpretação.

Foi necessária a utilização de um algoritmo especial no processo de comparação de *String's*. Gondim (2005) descreve alguns desses algoritmos como o *Levenshtein Metric*, *Smith Waterman*, *Jaro Metric*, *Soundex Distance Metric*, dentre outros. A opção escolhida foi o *Soundex Distance Metric* justamente pelas características relacionadas a desempenho e por ser o mais indicado na correção de erros fonéticos (GONDIM, 2005, p. 18). Utilizando o *Soundex* foi possível comparar as palavras-chave e o texto da solicitação e obter um índice de similaridade mais preciso. O sistema *Soundex* é um algoritmo fonético caracterizado por converter texto em um código *Soundex* baseado em fonemas. Os fonemas são as unidades básicas de uma língua e têm a propriedade de mudar de sentido e de uma palavra quando uma unidade é substituída por outra (FLANAGAN; Del Riesgo, 1990). Cada código *Soundex* consiste em uma letra e três números conforme agrupamento de letras com características fonéticas similares:

Assim, de acordo com esse conceito, a palavra “você” seria convertida ao código *Soundex* “V200”. A palavra “vc” também receberia o código “V200” obtendo-se assim uma similaridade entre as duas palavras. O código “V200” é formado da seguinte maneira:

- V: a inicial da palavra. No caso “você” e “vc”;
- 2: a segunda letra conforme Tabela 1. Note que a letra “o” não pertence a tabela portanto a mesma é ignorada, assim como a letra “ê”. Logo, tanto em “você” como “vc” a próxima letra a ser analisada é o “c”, e conforme a tabela recebe o número 2;
- 00: os outros dois números (dois zeros) completam os quatro dígitos que compõem o código *Soundex*.

A Figura 15 mostra o código da implementação de um simples teste de unidade com o objetivo de testar as comparações de *String*'s de maneira fonética utilizando-se o *Soundex*. Bem como, o resultado da execução do teste na Figura 16.



```

40  /**
41  * Test of comparacaoFonetica method, of class Soundex.
42  */
43  @Test
44  public void testComparacaoFonetica_String_String() {
45      System.out.println("Teste da comparacaoFonetica (Soundex)");
46      comparar("vc", "Você");
47      comparar("caqui", "caki");
48      comparar("sumir", "somir");
49      comparar("canto", "camto");
50      comparar("defenda", "dê fenda");
51      comparar("amigo", "amg");
52      comparar("beleza", "blz");
53      comparar("kely", "kelly");
54      comparar("catia", "cátia");
55      comparar("ma", "mau");
56      comparar("nmr", "numero");
57  }
58
59  private void comparar(String a, String b) {
60      boolean expectedResult = true;
61      boolean result = Soundex.comparacaoFonetica(a, b);
62      System.out.println(String.format("Comparando '%s' com '%s' = %s", a, b, result));
63      assertEquals(expectedResult, result);
64  }
65  }
66
67
68
69

```

Figura 15: Código da implementação do teste de unidade para o Algoritmo Soundex
Fonte: Autoria própria

5.3 ELABORAÇÃO DOS SERVIÇOS WEB

Os serviços Web foram então descritos segundo o BPMN, implementados em Java seguindo o padrão de metodos estabelecidos pela Interface 1 e publicados em um servidor *Web Apache Tomcat* utilizando o *Apache Axis2* (AXIS2, 2012) para a realização dos testes.

Uma vez implementadas, as classes *Java* foram compiladas, convertidas em serviços Web (Arquivos .aar) e publicados utilizando o *Netbeans* apoiado pelo arcabouço *Axis2 Plugin for Netbeans*. Uma vez publicados os serviços Web ficam disponíveis e podem ser explorados por outras aplicações ou serviços.

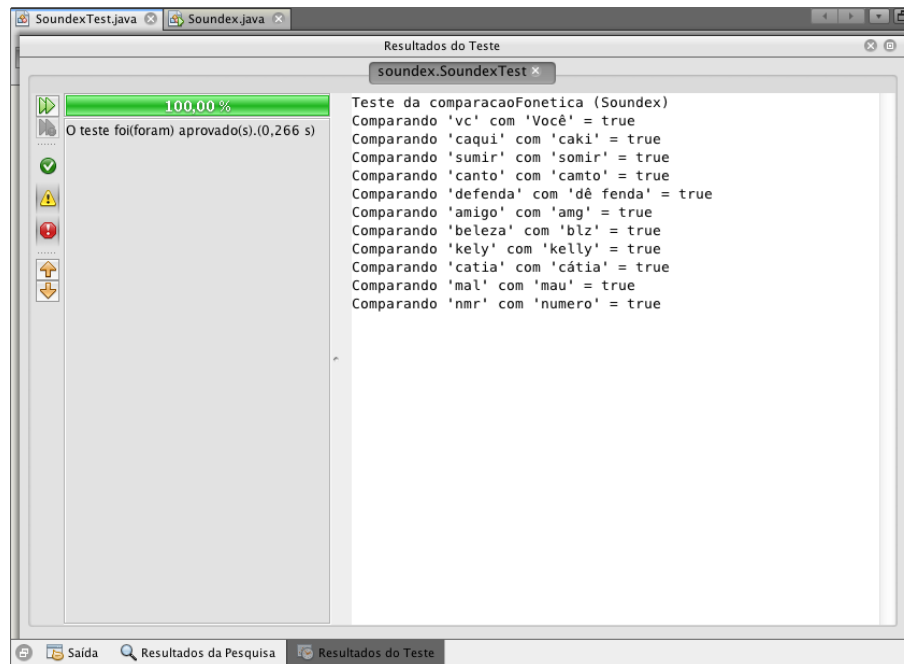


Figura 16: Resultado do teste de unidade para o Algoritmo Soundex
Fonte: Autoria própria

Interface 1 Interface em Java.

```
package br.edu.utfpr.cm.tsi.tcc.marcelo;
```

```
public interface WS {
    String getNameServiceRef();
    boolean handSheck(String orquestrador);
    String processar(String requisicao);
    String respostaLogica(String docxml);
}
```

5.3.1 SERVIÇOS WEB COM DADOS FICTÍCIOS

O termo *Mocks* se refere a objetos que simulam situações reais de forma controlada, com o objetivo de testar os diversos serviços que compõem a composição. Cada serviço desempenha de forma simulada ações que em um ambiente de produção ocorreria com dados reais oriundos das bases de dados dos sistemas acadêmicos. O ideal seria a utilização de tais dados, porém tal prática exige dissolução de todo um processo burocrático e a necessidade de autorização de instâncias superiores. Os dados para a persistência e criação dos mocks são oriundos de um banco de dados, inicialmente o MySQL, porém em decorrência de adversidades com o *Apache Tomcat*, o MySQL foi substituído pelo banco de dados orientados a objetos *Neodatis* que se mostrou satisfatório quanto ao desempenho e facilidade de manipulação e implantação.

5.3.2 MÉTODOS COMUNS NOS SERVIÇOS WEB

Os serviços Web desenvolvidos nesse trabalho possuem alguns métodos especiais conforme diagrama de classe na Figura 17.

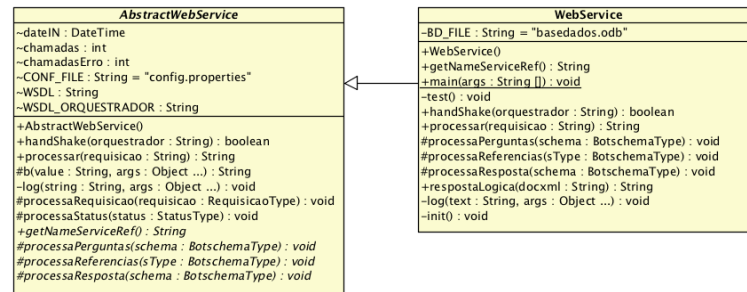


Figura 17: Diagrama de Classe que representa um Serviço Web

Fonte: Autoria própria

Os métodos *handShake* e *getNameServiceRef* estabelecem o papel de comunicação com o orquestrador. Enquanto o primeiro inicia o papel de reconhecimento e de teste o segundo retorna o nome que identifica ou referencia o serviço.

O método *processar* pode ser considerado como principal da classe, uma vez que o mesmo recebe como parâmetro uma *String* como parâmetro contendo o XML da solicitação e de acordo com o seu conteúdo gera uma resposta ao orquestrador, conforme o exemplo (Figura 18).

Em uma solicitação simples em XML em que observa-se a tag *<perguntas/>* vazia na Figura 18, esse é um indicativo de que o serviço Web deve retornar as perguntas ao orquestrador que posteriormente serão repassadas ao usuário com o qual mantém uma conversa.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <botschema>
3   <perguntas/>
4   <requisicao email="email@email.com">
5     <conteudo/>
6   </requisicao>
7   <process>true</process>
8 </botschema>
  
```

Figura 18: Requisição XML solicitando perguntas

Fonte: Autoria própria

A requisição XML é repassada ao método *“processar”* que retorna o mesmo XML porém com o conteúdo da tag *<perguntas>* preenchido. Conforme demonstrado pela Figura 19:

Esse mesmo método é utilizado para obter outras informações como referências (palavras-chave) e status do serviço.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <botschema>
3   <perguntas>
4     <pergunta regex="true">
5       <questao>pendências na biblioteca</questao>
6       <resposta>#biblioteca</resposta>
7       <messageErro>#nome, infelizmente não é possível continuar pois, você possui pendências na biblioteca. </messageErro>
8     </pergunta>
9     <pergunta regex="\w\d\d\d">
10      <questao>Qual sala deseja reservar? ex. F102</questao>
11      <messageErro>Informe a sigla da sala corretamente exemplo: F005 ou E103.</messageErro>
12    </pergunta>
13    <pergunta regex="^(?: (31) (\d) (0? [13578] | 1 [02] ) \2 | (29 | 30) (\d) (0? [13-9] |
14      1 [0-2] ) \5 | (0? [1-9] | \d | 2 [0-8] ) (\d) (0? [1-9] | 1 [0-2] ) \8 |
15      (?: 1 [6-9] | 2-9) \d | 7 \d (2) ) $ ^ (29) (\d) (0? 72) \12 (?: 1 [6-9] |
16      [2-9] \d) ? (?: 0 [48] | [2468] | [048] | [13579] [26] ) | (?: 16 | [2468]
17      [048] | [3579] [26] ) 00 $ ">
18      <questao>Qual data deseja efetuar a reserva? ex. 12/05/2013</questao>
19      <messageErro>Informe uma data válida (dd/mm/aaaa). Ex: 18/08/2013</messageErro>
20    </pergunta>
21    <pergunta regex="(0[1-9]|1[0-9]|2[0123])([.:])((012345)[0-9])">
22      <questao>Qual horário da reserva? ex. 16:35</questao>
23      <messageErro>Informe uma hora válida (hh:mm). Ex: 20:35</messageErro>
24    </pergunta>
25    <pergunta regex="\d+">
26      <questao>Qual duração da reserva em minutos? ex. 75</questao>
27      <messageErro>Informe apenas números. Ex: 60</messageErro>
28    </pergunta>
29    <pergunta>
30      <questao>email</questao>
31      <resposta>#email</resposta>
32    </pergunta>
33  </perguntas>
34  <requisicao email="marcelo@email.com">
35    <conteudo/>
36  </requisicao>
37  <process>true</process>
38 </botschema>

```

Figura 19: Resposta à requisição contendo as perguntas as serem respondidas

Fonte: Autoria própria

5.3.3 ESQUEMA XML

Conforme observado anteriormente, a manipulação da mensagem xml passada como parâmetro ao método “processar” do serviços Web é de suma importância. É imprescindível que todos os serviços reconheçam o padrão seguido pelo XML para que a sua manipulação seja possível. O diagrama da Figura 20 descreve o arquivo XML assim como detalhes técnicos de todas as suas ramificações.

5.4 ORQUESTADOR

Todos os serviços Web possuem um comportamento semelhante exceto o orquestrador de serviços. Esse serviço possui funções distintas que o tornam o componente principal desse trabalho. É ele quem recebe as solicitações oriundas dos usuários, interpreta e associa a solicitação a um ou mais serviços, extrai as dependências desses serviços (transformadas em perguntas) e estabelece uma conversa com o usuário até que o mesmo responda as perguntas. Após todas as respostas terem sido coletadas, o orquestrador invoca novamente o serviço Web para que ele processe a resposta da solicitação e repasse a mesma ao usuário, finalizando a conversa. Ao observar a Figura 19 notamos que, em algumas perguntas há a presença do parâmetro “regex”. Esse parâmetro, quando informado, é usado pra validar a resposta. Assim, não é preciso invocar o serviço Web pra validar a resposta, o que evita tráfego desnecessário de informação.

O orquestrador é o ponto principal desse trabalho. Cabendo ao mesmo a tarefa de coletar as requisições e através delas selecionar os serviços Web necessários para a obtenção

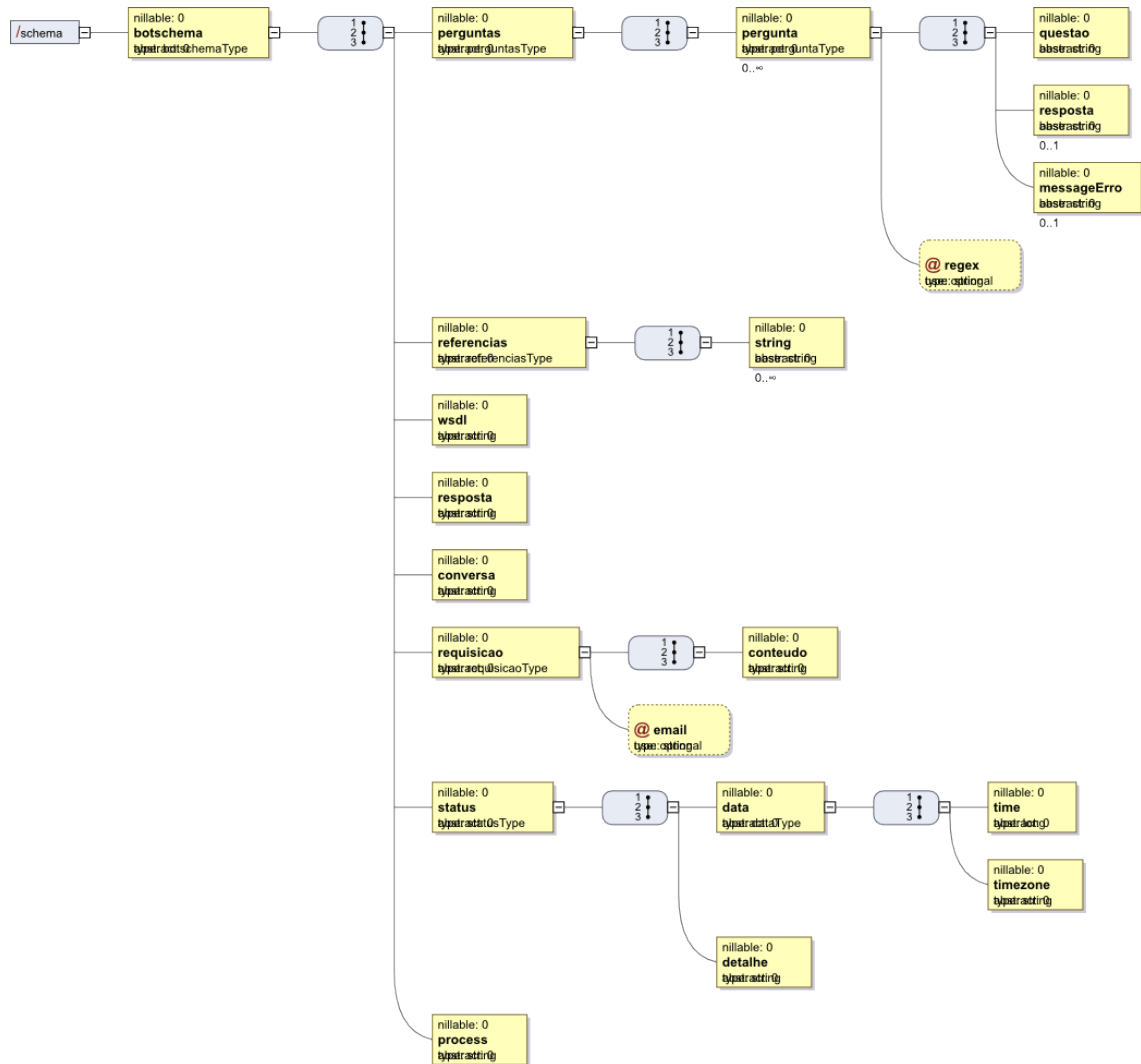


Figura 20: Diagrama que representa o arquivo XML para a comunicação com os serviços Web
Fonte: Autoria própria

da resposta que o usuário necessita. As vezes um serviço Web necessita de parâmetros para serem invocados e cabe ao orquestrador a tarefa de coletar esses parâmetros abstraídos por um questionamento com perguntas e respostas e repassá-los aos serviços Web. Cabe a ressalva de que o orquestrador esta preparado para, a qualquer momento receber novos serviços Web e agrega-los a composição de serviços sem a necessidade de parar o restante dos componentes ou serviços que compõem a orquestração.

O orquestrador possui alguns detalhes interessantes que merecem destaque, a começar pela presença de um arquivo XML denominado “servicos.xml”(Figura 21) onde estão registrados os endereços (*url's*) dos serviços Web que compõem a orquestração.

Ao ser inicializado o orquestrador efetua a leitura do arquivo “serviços.xml” e coleta os endereços dos serviços Web registrados. Posteriormente, inicia um processo de conexão (“linkagem”) com cada serviço listado. Obedecendo aos seguintes passos:

```

<sevidos urlOrquestrador=" http://localhost:8080/axis2/services/Orquestrador2Service">
  <WebServices>
    <string>http://localhost:8080/axis2/services/WebServiceNotasPorAluno?wsdl</string>
    <string>http://localhost:8080/axis2/services/WebServiceDadosUsuarios?wsdl</string>
    <string>http://localhost:8080/axis2/services/WebServiceSaudacao?wsdl</string>
    <string>http://localhost:8080/axis2/services/WebServiceBiblioteca?wsdl</string>
    <string>http://localhost:8080/axis2/services/WebServiceReservaSala?wsdl</string>
    <string>http://localhost:8080/axis2/services/WebServicePesquisaUsuarios?wsdl</string>
    <string>http://localhost:8080/axis2/services/WebServiceCadastrarUsuarios?wsdl</string>
    <string>http://localhost:8080/axis2/services/WebServiceMatricula?wsdl</string>
    <string>http://localhost:8080/axis2/services/WebServiceCancelarReservaSala?wsdl</string>
  </WebServices>
</sevidos>

```

Figura 21: Código XML presente no arquivo “servicos.xml”

Fonte: Autoria própria

- Verifica se o serviço está disponível;
- Invoca o método *handShake* dos serviços Web cuja resposta esperada é uma *String* com o conteúdo “true”;

Qualquer negativa aos passos acima, a tentativa de conexão com o serviço Web é abortada e posteriormente o orquestrador efetuará uma nova tentativa de conexão a cada dois minutos até que obtenha sucesso e todos os serviços Web sejam verificados.

Paralelamente o orquestrador captura a *hash MD5* (Classe 1) do arquivo “servicos.xml” no momento da execução e o armazena em uma variável local. A cada dois segundos uma nova captura é realizada e comparada com a variável local com o objetivo de detectar se o arquivo foi alterado. Caso seja detectada alguma alteração caracteriza-se um indicativo de que a lista de serviços Web foi alterada com a inclusão ou remoção de novos endereços de serviços e um novo processo de conexão é iniciado.

Classe 1 Classe Java que verifica se houve alteração no arquivo “*servicos.xml*”.

```

class ReminderTask extends TimerTask {
    @Override
    public void run() {
        String md5now = getMD5FromFile(file);
        if (!md5.equals(md5now)) {
            B.log(md5.isEmpty()
                ? "Extraindo hash do arquivo {0}\n{2} "
                : "Arquivo {0} foi alterado em {1}",
                file.getName(),
                new Date(),
                md5now);
            md5 = md5now;
            fireChangeFile(file);
        }
    }
}

```

5.4.1 PREPARAÇÃO DO ORQUESTRADOR

Para que o orquestrado possa associar as solicitações a um determinado serviço é preciso alimentá-lo com referencias (palavras-chave, nome dos serviços, lista de perguntas, *status* do serviço, dentre outra) oriundas dos serviços Web. Para isso ao inicializar e consolidar todas as conexões, há uma chamada ao método *processar* dos serviços Web passando como parâmetro um XML com a tag *<referencias>* vazia, obtendo como resposta o mesmo XML porém com a tag preenchida. Conforme mencionado anteriormente essa é a tática utilizada para obter informações referentes a cada serviço. Após, essa etapa o orquestrados finalmente entra totalmente em operação aguardando as solicitações.

5.4.2 INTERPRETAÇÃO DAS SOLICITAÇÕES

O orquestrado possui um método principal denominado *conversa*, que é a porta de entrada para a comunicação entre a orquestração e os meios externos. Esse método recebe como parâmetro um XML padronizado que traz, preenchidas as tags “*<email>*“ e “*<conteudo>*”. Enquanto a primeira se encarrega de identificar quem acessa a orquestração a segunda é a mensagem em si, que posteriormente será interpretada pelo orquestrador. De posse da solicitação a primeira ação é verificar se já há uma conversa em andamento. Se houver o orquestrador interpreta que o usuário está respondendo a alguma questão e seta a resposta à pergunta não respondida da vez, do contrario o orquestrador interpreta que ainda não há uma conversa em andamento e inicia uma nova, começando por tentar associar um serviço à solicitação.

O processo de associação da solicitação aos serviços registrados, inicia-se com a segregação da solicitação em fragmentos de palavras. Os fragmentos são comparados por intermédio do *Soundex* com as palavras-chave extraídas dos serviços e a cada assertiva é atribuído um ponto ao serviço Web avaliado; O serviço Web que obter maior pontuação ou maior numero de assertivas é selecionado; Se houver empate ou nenhum serviço receba assertivas o orquestrador solicita maior explicitude por parte do solicitante. Uma vez selecionado um serviço Web uma conversa é estabelecida.

5.4.3 CONVERSA

A relação entre usuário é orquestração resume-se em perguntas e respostas. Ao haver uma conversa estabelecida (Figura 21a) a primeira atitude é por parte do orquestrador que extrai as perguntas (quando houver) do serviço e repassa uma a uma na forma de uma questão (Figura 21b). Uma vez respondida o orquestrado seta a resposta a pergunta selecionada e repassa a

próxima até que todas as perguntas sejam respondidas. Uma vez respondidas o orquestrador invoca o serviço Web selecionado, passando por parametro as perguntas respondidas e extrai a resposta final repassando ao usuário e encerra a conversa.

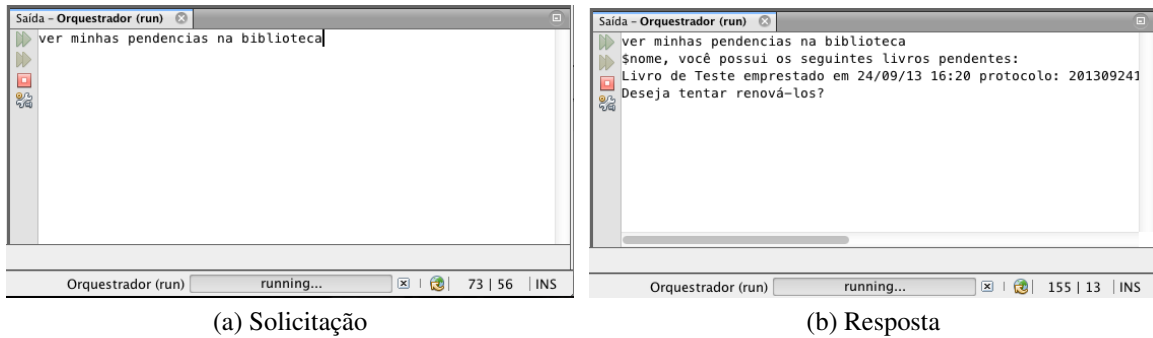


Figura 22: Conversa entre o solicitante e a orquestração

Fonte: Autoria própria

6 CONCLUSÃO

Nesse trabalho foram discutidos conceitos referentes ao uso de serviços Web em âmbito institucional, unido às redes sociais e comunicadores instantâneos. Esses conceitos demonstram grande potencial de melhoria na usabilidade de tarefas cotidianas. Ademais, o uso de serviços Web pode amenizar esforços de programação, potencializar recursos, quebrar paradigmas e encurtar distâncias entre diferentes sistemas, setores ou segmentos. Soma-se a esse potencial, questões como composição de serviços – nesse trabalho, a orquestração de serviços – e temos uma ferramenta totalmente extensível.

Pensando nesses benefícios, foi proposta uma ferramenta para uso na UTFPR-CM, com intuito de simular tarefas, criando um projeto piloto que podem vir a ser implantado no futuro. A forma de consumir as informações geradas pela orquestração de serviços apresentada nesse trabalho (uso de comunicadores instantâneos) demonstra ser uma ideia interessante, tendo em vista que, vai além da simples construção de um formulário, páginas Web ou outros tipos de mídia usualmente ofertadas por sistemas similares e aproxima-se mais de uma comunicação formal, similar a uma conversa.

O uso de internet cresce cada vez mais, e por consequência, as redes sociais estão cada vez mais presentes no cotidiano das pessoas, assim como a demanda por informações. Atrair informações e serviços às redes sociais desponta como um importante conceito. Agrega-se a isto o poder de uma rede de serviços Web cuja principal característica é a busca por informações e o resultado é uma ferramenta com potencial institucional muito interessante. Dessa forma, a ideia apresentada nesse trabalho contribui para a formação de um elo informativo e interativo entre usuários e instituição.

6.1 TRABALHOS FUTUROS

A filosofia com a qual esse projeto foi concebido, é voltada a questões como expansibilidade, interdependência e modularização, conceitos que possibilitam uma evolução contínua das funcionalidades existentes e a fácil inclusão de novas funcionalidades proporcionando uma

experiência ainda mais ampla e melhorada. O atendimento correto e de qualidade à uma eventual solicitação está diretamente ligada a quantidade de serviços Web vinculados. Em contrapartida a agregação de novos serviços de forma descentralizada exige um esforço extra quanto planejamento, controle e tratamento de eventuais exceções.

Nesse sentido, é interessante deixar uma lista com sugestões de melhorias que podem vir a auxiliar usuários continuidade desse projeto:

- Implantar planejadores JSHOP2 no auxílio ao gerenciamento e inserção de novos serviços Web, haja visto que, essa ferramenta auxilia na organização da composição a medida que novos serviços Web são agregados;
- Mapear a orquestração dos serviços através da anotação WSBPEL;
- Estudar as questões de acessibilidade de forma a adaptar o sistema para o uso por deficientes visuais, tendo em visto a importância de tal abordagem em um ambiente acadêmico.

REFERÊNCIAS

- AXIS2, T. A. S. F. **Apache Axis2 - Next Generation Web Services**. 04 2012. Disponível em: <http://axis.apache.org/axis2/java/core/>. Acesso em: 5 out. 2013, 13:12.
- BASTIEN, J. M. C.; SCAPIN, D. L.; LEULIER, C. The ergonomic criteria and the iso/dis 9241-10 dialogue principles: a pilot comparison in an evaluation task. **Interacting with Computers**, v. 11, n. 3, p. 299–322, 1999. Disponível em: <http://dblp.uni-trier.de/db/journals/iwc/iwc11.html#BastienSL99>.
- BRANDÃO, R. C. **Genesis System: Proposta de uma Arquitetura Conceitual para Orquestração de Serviços em Ambientes Autônômicos Orientados a Serviço**. Tese (Doutorado) — Instituto Nacional de Telecomunicações, 2012.
- CARR, R. **The Soundex Algorithm**. 2010. Disponível em: <http://www.blackwasp.co.uk/Soundex.aspx>. Acesso em: 27 out. 2011, 14:55.
- CHRISTENSEN, E.; CURBERA, F.; MEREDITH, G.; WEERAWARANA, S. **Web Services Description Language (WSDL) 1.1**. 2001. W3C Note 15 March 2001. Disponível em: <http://www.w3.org/TR/wsdl>. Acesso em: 13 set. 2013, 16:03.
- COLLIER, R. **Smack API**. 2013. Disponível em: <http://www.igniterealtime.org/projects/smack>. Acesso em: 20 jun. 2013, 13:57.
- DAMASCENO, J. C. **Introdução à Composição de Serviços Web**. [S.l.], 2009. Disponível em: <http://www.ufpi.br/subsiteFiles/ercemapi/arquivos/files/minicurso/smc8.pdf>.
- ERL, T. **Service-Oriented Architecture: Concepts, Technology, and Design**. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2005. ISBN 0131858580.
- FIELDING, R.; GETTYS, J.; MOGUL, J.; FRYSTYK, H.; MASINTER, L.; LEACH, P.; BERNERS-LEE, T. **Hypertext Transfer Protocol – HTTP/1.1**. United States: RFC Editor, 1999.
- FLANAGAN, J. L.; Del Riesgo, C. J. Speech processing: a perspective on the science and its applications. **ATT-TECH**, v. 69, n. 5, p. 2–13, set./out. 1990. ISSN 8756-2324.
- GONDIM, F. M. **Algoritmo de Comparação de Strings para Integração de Esquemas de Dados**. 2005. Disponível em: <http://www.cin.ufpe.br/~tg/2005-2/fmg.pdf>. Acesso em: 4 out. 2013, 23:12.
- INSITE. **Converse com o Robô Ed!** 2013. Disponível em: <http://www.ed.conpet.gov.br/br/converse.php>. Acesso em: 8 ago. 2013, 21:39.
- KOMESU, F. Internetês para interneteiros:(velhas) questões sobre escrita. **Estudos Linguísticos**, v. 1, p. 1000–1007, 2007.

- LEOV, B. **Java MSN API**. 2008. Disponível em: <http://sourceforge.net/projects/robomsn/>.
- MOUTA, A. E. B. **Manual de Implantação da Nota Fiscal Eletrônica**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2010. ISBN 9788573939538.
- OMG, O. M. G. I. **BPMN Information Home**. 2011. Disponível em: <http://www.bpmn.org>.
- PAPAZOGLU, M. P.; TRAVERSO, P.; RICERCA, I.; TECNOLOGICA, S. Service-oriented computing: State of the art and research challenges. **IEEE Computer**, v. 40, p. 2007, 2007.
- PELTZ, C. Web services orchestration and choreography. **Computer**, v. 36, n. 10, p. 46–52, 2003. ISSN 0018-9162.
- QUIN, L. R. E. **XML Essenciais**. 2010. Online. W3C, XML Technology. Acessado em 12/08/2013. Disponível em: <http://www.w3.org/standards/xml/core>. Acesso em: 12 ago. 2010, 17:45.
- SANTOS, R. C. D. **Desenvolvimento de uma metodologia para avaliação de usabilidade de sistemas utilizando a lógica Fuzzy baseado na ISO**. Tese (Doutorado) — Faculdade De Economia E Finanças Ibmec Programa de Pós-Graduação e Pesquisa em Administração e Economia, Rio de Janeiro, 2007.
- SILVA, A. N. d. **Composição Automática de Web Services: Avaliando os planejadores JSHOP2 e JESS**. Tese (Doutorado) — Universidade Federal da Bahia, 2007. Disponível em: <http://homes.dcc.ufba.br/~dclaro/download/Monografia%20-%20Final%20II%20-%20AlexandroNascimento.pdf>. Acesso em: 16 set. 2013, 18:32.
- TOMAÉL, M. I.; ALCARÁ, A. R.; CHIARA, I. G. D. Das redes sociais à inovação. **Ciência da informação, Brasília**, SciELO Brasil, v. 34, n. 2, p. 93–104, 2005.
- TOMCAT, T. A. S. F. **Apache Tomcat**. 2013. Disponível em: <http://tomcat.apache.org>. Acesso em: 29 set. 2013, 12:00.
- VICTORIANO, A. P. D.; CARDOSO, G. A. G. **Web Services Choreography**. 01 2010. Universidade de São Paulo. <Http://ccsl.ime.usp.br/wiki/images/7/79/Choreographies.pdf>.
- W3C. **Web of Services - W3C**. 2013. Disponível em: <http://www.w3.org/standards/webofservices/>. Acesso em: 12 set. 2013, 22:01.