

UNIVERSIDADE FEDERAL DE ALFENAS
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Alberto Silveira Carilo
Gabriela Moreira da Silva

**EVENTOS UNIFAL:
UM APLICATIVO PARA OBTER INFORMAÇÕES ACERCA DE
AULAS E DEMAIS EVENTOS DA UNIFAL-MG ATRAVÉS DE
COMANDOS DE VOZ**

Alfenas, 06 de Julho de 2015.

UNIVERSIDADE FEDERAL DE ALFENAS
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**EVENTOS UNIFAL:
UM APLICATIVO PARA OBTER INFORMAÇÕES ACERCA DE
AULAS E DEMAIS EVENTOS DA UNIFAL-MG ATRAVÉS DE
COMANDOS DE VOZ**

Alberto Silveira Carilo
Gabriela Moreira da Silva

Monografia apresentada ao Curso de Bacharelado em
Ciência da Computação da Universidade Federal de
Alfenas como requisito parcial para obtenção do Título de
Bacharel em Ciência da Computação.

Orientadores: Prof. Luiz Eduardo da Silva
Prof. Flavio Barbieri Gonzaga

Alfenas, 06 de Julho de 2015.

Alberto Silveira Carilo
Gabriela Moreira da Silva

**EVENTOS UNIFAL:
UM APLICATIVO PARA OBTER INFORMAÇÕES ACERCA DE
AULAS E DEMAIS EVENTOS DA UNIFAL-MG ATRAVÉS DE
COMANDOS DE VOZ**

A Banca examinadora abaixo-assinada aprova a monografia apresentada como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação pela Universidade Federal de Alfenas.

Prof. Luiz Eduardo da Silva (Orientador)

Universidade Federal de Alfenas

Prof. Flavio Barbieri Gonzaga (Orientador)

Universidade Federal de Alfenas

Prof. Leonardo Aparecido Ciscon

Universidade Federal de Alfenas

Profa. Leandra de Carvalho Nogueira

Universidade Federal de Alfenas

Alfenas, 06 de Julho de 2015.

[Dedicamos este trabalho a Deus, aos nossos Pais, familiares, amigos e orientadores.]

AGRADECIMENTO

Agradecemos a Deus, que nos guiou e protegeu a todo o tempo, para a conclusão desta etapa; aos nossos pais, o nosso muito obrigado pela oportunidade e apoio concedido; aos nossos familiares e aos nossos amigos pela amizade e companheirismo.

Agradecemos também aos nossos orientadores, que tanto se fizeram presentes no desenvolvimento deste trabalho.

“Se eu vi mais longe foi por estar sobre os ombros de gigantes.”

Isaac Newton

RESUMO

A disseminação de ferramentas comandadas via entrada de voz é crescente e muito notável atualmente. Devido à sua comodidade e à naturalidade em que se torna a interação humano-máquina, o comando de voz se faz presente em diversos dispositivos eletrônicos. Este trabalho demonstra o desenvolvimento de um sistema exclusivo para a plataforma Android, que permite a consulta de informações acerca de aulas e demais eventos da Universidade Federal de Alfenas - Minas Gerais (UNIFAL-MG), através de um aplicativo que permite a busca destas informações por entrada de voz. A intenção deste sistema é agilizar o acesso às informações de aulas e de eventos da universidade, como horários e localizações, que atualmente estão no sistema de gerenciamento de eventos “Porteiro Web”, disponibilizando-as para consulta simples, rápida e com o mínimo de tráfego de dados possível, facilitando e agilizando parte do dia-a-dia das pessoas que frequentam e usufruem dos serviços da universidade. |

Palavras-Chave: |Android, Busca por voz, Reconhecimento de voz, Usabilidade, Web Crawler, Web Service. |

ABSTRACT

The dissemination of tools controlled by voice input is growing and currently outstanding. Because of its convenience and ease of becoming a human-machine interaction, the voice command is present in many electronic devices. This work demonstrates the development of an application for Android platform, that allows you to search information about classes and other events of the Federal University of Alfenas - Minas Gerais (UNIFAL-MG), through voice input. The intent of this system is simplify the access to classes information and to events of the university, such as schedules and locations, which are currently in the event management system "Porteiro Web", making them available for simple, and easy access, with minimal possible of data traffic, simplifying and speeding up the daily life of the people who use and enjoy the services of the university. |

Keywords: | Android, Search by voice, Speech Recognize, Usability, Web Crawler, Web Service. |

LISTA DE FIGURAS

FIGURA 1 - TESTE DE COMPATIBILIDADE COM DISPOSITIVOS MÓVEIS.....	25
FIGURA 2 - SISTEMA PORTEIRO WEB	26
FIGURA 3 - COMPARATIVO ENTRE WEB SERVICE RESTFUL E SOAP	33
FIGURA 4 - TRECHO EM JSON COM OBJETOS QUE CONSTITUEM A LISTA DE EVENTOS.....	34
FIGURA 5 - ARQUITETURA UTILIZADA NO PROJETO	37
FIGURA 6 - DUMP DISPONIBILIZADO PELA WIKIPEDIA.....	39
FIGURA 7 - PADRÃO DAS URLS DO SISTEMA PORTEIRO WEB.....	40
FIGURA 8 - PÁGINA COM A DESCRIÇÃO COMPLETA DE UM EVENTO NO PORTEIRO WEB.....	41
FIGURA 9 - CÓDIGO FONTE DA PÁGINA DE DESCRIÇÃO DE UM EVENTO NO PORTEIRO WEB.....	42
FIGURA 10 - ESBOÇO UTILIZADO NA ELABORAÇÃO DAS POSSÍVEIS PERGUNTAS	43
FIGURA 11 - ILUSTRAÇÃO DA PADRONIZAÇÃO	46
FIGURA 12 - EXEMPLO DE QUERY SQL UTILIZANDO O OPERADOR LIKE E O SÍMBOLO %	48
FIGURA 13 - TELA EXIBINDO EVENTOS E MOSTRANDO AS RECORRENTES ABREVIATURAS	49
FIGURA 14 - QUERY SQL COM RESULTADO VAZIO DEVIDO AO PROBLEMA DAS ABREVIATURAS	50
FIGURA 15 - COMPARAÇÃO DAS STRINGS WRITERS E VINTNER UTILIZANDO ALINHAMENTO GLOBAL..	51
FIGURA 16 - PROTÓTIPO INICIAL DA TELA DO APLICATIVO	55
FIGURA 17 - TELA DO APLICATIVO, COM DESTAQUE AOS COMPONENTES DE LAYOUT	57
FIGURA 18 - EXEMPLO DE URL PARA CONSULTA AO WEB SERVICE	57
FIGURA 19 - RESULTADO DO WEB SERVICE E DA TELA DO APLICATIVO PARA UMA PERGUNTA ESPECÍFICA	58

LISTA DE TABELAS

TABELA 1 - ALINHAMENTO POSSÍVEL I.....	52
TABELA 2 - ALINHAMENTO POSSÍVEL II.....	52
TABELA 3 - ALINHAMENTO POSSÍVEL III.....	52
TABELA 4 - ALINHAMENTO MÁXIMO.....	53
TABELA 5 - TAMANHO DO TÍTULO X TEMPO DE RESPOSTA	60

LISTA DE ABREVIACÕES

DOM	Documento Object Model
JSON	JavaScript Object Notation
RU	Restaurante Universitário
SQL	Structured Query Language
UNIFAL-MG	Universidade Federal de Alfenas
URIs	Uniform Resource Identifiers
URL	Uniform Resource Locator
XML	eXtensible Markup Language

SUMÁRIO

1 INTRODUÇÃO	23
1.1 JUSTIFICATIVA E MOTIVAÇÃO	24
1.2 PROBLEMATIZAÇÃO	25
1.3 OBJETIVOS	26
1.3.1 Gerais	26
1.3.2 Específicos	27
2 REVISÃO BIBLIOGRÁFICA	29
3 ASPECTOS TECNOLÓGICOS	31
3.1 WEB CRAWLER	31
3.1.1 Python	31
3.1.1.1 Scrapy	32
3.2 WEB SERVICE	32
3.2.1 Web Service RESTful (REST)	33
3.2.2 Arquivo JSON	34
3.3 ANDROID	35
3.3.1 Speech Recognizer	35
4 FUNDAMENTAÇÃO TEÓRICA E METODOLOGIA	37
4.1 ARQUITETURA DO SISTEMA	37
4.2 ESTUDO DO PORTEIRO WEB	38
4.3 DESENVOLVIMENTO DO WEB CRAWLER	40
4.4 ELABORAÇÃO DAS PERGUNTAS	43
4.5 EXTRAINDO DADOS DAS PERGUNTAS	44
4.5.1 Stop Words	44
4.5.2 Capturando dados para a busca	45
4.6 OBTENDO OS RESULTADOS A PARTIR DO BANCO DE DADOS	47
4.6.1 MySQL – Operações utilizando o operador “LIKE”	47
4.7 COMPLICAÇÕES	49
4.7.1 Alinhamento de Strings	50
4.7.1.1 Alinhamento Global	51
4.7.1.2 Alinhamento Local	53
4.7.1.3 Adaptações ao algoritmo de Alinhamento Local e atribuição de pesos	54
4.8 DESENVOLVIMENTO DO APLICATIVO	55
4.8.1 Protótipo	55
4.8.2 Desenvolvimento Android	56
5 RESULTADOS	59
6 CONCLUSÃO	61
7 TRABALHOS FUTUROS	63
7.1 RESTAURANTE UNIVERSITÁRIO (RU)	63
7.2 PADRONIZAÇÃO DO SISTEMA “PORTEIRO WEB”	63
7.3 EXPANSÃO DO APLICATIVO	64
8 REFERÊNCIAS BIBLIOGRÁFICAS	65

|

1

Introdução

Este capítulo apresenta alguns detalhes sobre a problemática, a criação do sistema Eventos Unifal, sua justificativa, objetivos e motivação para tal.

No cotidiano universitário existe a necessidade de se obter informações de localizações ou de horários tanto de aulas como de eventos, sendo preciso, muitas das vezes, acessar os sistemas da universidade que disponibilizam tais informações, ou até mesmo recorrer a um professor ou responsável para que o mesmo consulte a informação desejada no sistema e a retorne ao aluno.

Além de considerar o problema anteriormente descrito, sabe-se que operar um dispositivo através de comandos voz torna o uso muito mais intuitivo e promove certa liberdade ao usuário, podendo manipular o sistema estando com as mãos livres para realizar outras tarefas; ou seja, o uso da voz para controlar um sistema agrega muito mais usabilidade ao mesmo. Devido a estes benefícios, há um grande desejo e esforço por parte das empresas de desenvolvimento de software para incrementar este tipo de função em seus produtos.

Reconhecedores automáticos de voz automáticos tem sido desenvolvidos há pelo menos 30 anos, porém apenas a partir da metade dos anos 90 alcançaram maturidade suficiente para uso em aplicações reais (FERNANDES, 2004).

A fim de agilizar o cotidiano dos universitários, este projeto descreve a criação de uma ferramenta capaz de trazer informações sobre aulas e demais eventos da UNIFAL-MG em um aplicativo, tais como datas, horários e localização, além de informações úteis como: todas as aulas ou eventos que serão ministrados no dia por alguma pessoa informada; tudo de forma rápida e simples, através de comandos de voz do usuário. |

1.1 Justificativa e Motivação

Um dos motivos que justifica o desenvolvimento e o crescimento da tecnologia é o ganho de tempo que ela promove. Deve-se então, pensar concomitantemente: na interação do homem com a máquina.

A comunicação oral é, sem dúvida, a forma mais natural de comunicação humana. Em virtude da interação humano-computador se tornar cada vez mais comum, surge uma demanda natural por sistemas capazes de interagir com os usuários de maneira mais prática e intuitiva. (MACIEL, 2012)

As informações de aulas e demais eventos que o estudante necessita, normalmente, se encontram nos meios da universidade (sistema, murais, etc) e às vezes o acesso a estes se torna dispendioso devido à sua localização física ou devido à dificuldade em ter que acessar o site do sistema e navegar através do mesmo até poder, então, efetuar uma busca manual da informação desejada. Além disto, o site¹ em questão não é responsivo², conforme mostra a Figura 1, onde se encontra o resultado de um teste de compatibilidade com dispositivos móveis feito através de uma ferramenta disponibilizada pelo Google (TESTE GOOGLE, 2015), piorando assim a usabilidade, considerando o uso via smartphone.

Unindo a agilidade em utilizar uma busca por voz, com a praticidade de um aplicativo móvel, é possível criar uma ferramenta de auxílio ao cotidiano da comunidade da UNIFAL-MG.

Porteiro Web¹: <https://www.unifal-mg.edu.br/app/graduacao/>

Responsivo²: técnica de estruturação que permite que o site se adapte ao browser do usuário.

Teste de compatibilidade com dispositivos móveis 8+1

https://www.unifal-mg.edu.br/app/graduacao/porteiroweb/day.php?day=08&month=06&year=2015&area=3&room=4 ANALISAR

Não compatível com dispositivos móveis

A página não parece ser compatível com dispositivos móveis

- ✗ Texto muito pequeno para ler
- ✗ Links muito próximos
- ✗ Viewport não configurada para dispositivo móvel

Para ver detalhes sobre quais partes da página são afetadas pelos problemas de usabilidade, consulte [Pagespeed Insights](#).

Como o Googlebot vê esta página



Tornar esta página compatível com dispositivos móveis

Escolha a opção que melhor descreve a maneira que você criou este site:

Usei CMS (sistema de gerenciamento de conteúdo)
Utilizei um software como o WordPress ou o Joomla.
Próxima

Outra pessoa criou este site para mim
Contratei uma pessoa para criar este site e gostaria de conselhos sobre como trabalhar com um desenvolvedor.
Próxima

Eu criei este site
Eu criei este site e sei programar.
Próxima

Você usa as Ferramentas do Google para webmasters?

Figura 1 - Teste de compatibilidade com dispositivos móveis

1.2 Problematização

O sistema de gerenciamento de eventos da UNIFAL-MG, denominado “Porteiro Web”, contém informações úteis aos docentes e discentes, porém, seu uso é notavelmente mais frequente entre os docentes e funcionários.

Além da não popularidade do sistema entre os discentes, a utilização do mesmo para obter informações de forma rápida e prática é insatisfatória, devido a necessidade de se ter em mãos um notebook, para que se possa acessar o site do sistema, navegar entre as várias páginas do mesmo, até que se possa fazer uma busca manual e finalmente, obter a informação desejada.

Uma das telas do sistema “Porteiro Web”, contendo registros de eventos, é mostrada na Figura 2.

Porteiro Web - Agendamento de Salas

1 Jul 2015 Ver Agenda Sistema Bloqueado para administração Pesquisa Relatório Usuário Desconhecido

Unifal-MG Avaliação Central de Aulas Alterar

quarta 01 julho 2015

1 para dia anterior 8 para hoje 8 para dia seguinte

Solicitar manutenção de computadores (NTI) Solicitar manutenção presencial de projetor multimídia (D5G)

Horário	SAB-PCIC-001 (Cab. 40)	SAB-PCIC-002 (Cab. 40)	SAB-PCIC-003 (Cab. 53)	SAB-PCIC-004 (Cab. 53)	SAB-PCIC-005 (Cab. 70)	SAB-PCIC-006 (Cab. 54)	SAB-PCIC-007 (Cab. 42)	SAB-PCIC-008 (Cab. 70)	SAB-PCIC-009 (Cab. 40)	SAB-PCIC-010 (Cab. 70)
07:00	CONCURSO FRENTE GERAL N° 1032014 - RESERVA DE DISCIPLINA E BAIXA INTEGRADA		ENF X AMB		RAUDIO	PARA BTO CL	AULA DO PPGEN		FIBRO	
08:00									MICRO	SEM - MEDICINA
09:00			OP UNIT	INSTRUM	CALCULO I	GEOMORFO		BIO CONSERV	BIOGUM	
10:00										
11:00						NUT BA SCA				
12:00										
13:00		REC VAC	ALTO		ESTATISTICA	BIO CEL	AULA FARMOP	IMUNOLOGIA	BIOGUM	FARM HOSP
14:00				BIO CEL						
15:00		GESTÃO	SEMINARIOS FARMACIA-ESCOLA	INSTR	RAUDIO	HEMATO CL	PROB HISTORIA	QUIM GERAL II	GEN ANALITICA	PROVA DE QUIMICA GERAL EXPERIMENTAL
16:00										
17:00	REUNIO SEMANA DA BIOLOGIA	OPATIVA VIROLOGIA CLINICA	MONITORIA ESTATISTICA		REUNIAO DO GRUPO DE CENTALOGRAFIA	MONITORIA FARM, NUT E FIBRO				
18:00										REUNIAO ESTAGIO
19:00										
20:00	MAT SUP		INT CON FIS	CALCULO I	LIT AMIGA I	BIOGUM	MET FISICA II	FUND EDUC	FIS GUM I	
21:00										SOU
22:00										
23:00										
24:00										
25:00										
26:00										
27:00										

Figura 2 - Sistema Porteiro Web

Vale ressaltar ainda, que o sistema não é responsivo, dificultando o uso via smartphone, lugar onde justamente a agilidade e a rapidez da interação do usuário com o dispositivo, determinam a viabilidade do sistema. Além do tempo gasto percorrendo as etapas descritas anteriormente, ocorre um tráfego de dados considerável durante a navegação no sistema, o que é absolutamente indesejável em um ambiente de computação móvel.

Visando melhorar este cenário, surge o seguinte problema: É possível criar um sistema comandado por voz, que visa trazer informações de uso cotidiano, contidas no sistema “Porteiro Web”, de forma fácil, rápida e intuitiva? |

1.3 Objetivos

1.3.1 Gerais

O objetivo deste trabalho é prover acesso às informações acerca de aulas e demais eventos da UNIFAL-MG, presentes no sistema de gerenciamento de eventos “Porteiro Web”, de forma rápida, ágil e intuitiva àqueles que frequentam e usufruem dos serviços da universidade, sem precisar navegar através do sistema, consumindo tempo e tráfego de dados desnecessários. |

1.3.2 Específicos

Para alcançar o sucesso no projeto, cumprindo o proposto nos objetivos gerais, será preciso realizar uma lista específica de objetivos:

- Planejamento geral do projeto.
- Estudo da linguagem de programação Python (seção 3.1.1).
- Desenvolvimento do Web Crawler (seção 3.1).
- Elaboração de possíveis e frequentes perguntas do usuário para o aplicativo.
- Desenvolvimento do Webservice (seção 3.2), já trazendo resultados da busca.
- Estudo da plataforma de desenvolvimento Android (seção 3.3).
- Desenvolvimento do aplicativo Android.
- Testes e avaliação do aplicativo desenvolvido

|
|

2

Revisão Bibliográfica

Este capítulo apresenta uma revisão bibliográfica básica sobre usabilidade de uma forma geral, Android, reconhecimento de voz e seus benefícios no contexto do auxílio ao cotidiano.

Analisando alguns trabalhos e artigos, percebeu-se a gama de benefícios que se tem através do uso de entradas de voz em aplicações móveis, bem como a importância da usabilidade de sistemas e de aplicativos móveis.

ISHIBASHI (2014), em seu trabalho, descreve a criação do layout de um aplicativo de cunho acadêmico, também destinado à consulta de informações contidas em um sistema já existente, onde foi dado destaque à presença do smartphone atualmente: “Hoje a participação do smartphone na vida das pessoas vem crescendo, e com isso vem crescendo também o número de usuários que acessa a Internet por meio de dispositivos móveis.”

Além disso, também no trabalho de ISHIBASHI (2014), nota-se uma grande preocupação em relação à usabilidade, fato que motivou o seu trabalho, onde mesmo na existência de um sistema destinado à informações acadêmicas, houve interesse de sua parte em criar um aplicativo móvel destinado à exibição organizada e agradável das informações.

Apesar do [Sistema Integrado de Informações Acadêmicas] estar disponível aos acadêmicos, o que se percebeu [...] é que muitas informações [...] foram apenas disponibilizadas, de maneira desorganizada, sem preocupação estética, o que pode confundir o acadêmico e desestimulá-lo a consultar regularmente o Sistema [...] Em se tratando de dispositivos móveis, a apresentação visual é uma questão fundamental (ISHIBASHI, 2014)

No trabalho de VERTANEN e KRISTENSSON (2009), da Universidade de Cambridge, foi desenvolvido um aplicativo denominado Parakeet, para reconhecimento de voz utilizando Pocketsphinx (versão mobile da ferramenta de reconhecimento de voz Sphinx¹), reconhecendo a fala para os idiomas Inglês

Americano e Inglês Britânico; sendo desenvolvido para dispositivos móveis baseados em Linux.

No projeto do aplicativo Parakeet, foram realizados experimentos computacionais e com usuários, onde os participantes tiveram uma taxa média de entrada de texto de 18 palavras por minuto no interior de suas casas e 13 palavras por minuto durante uma caminhada ao ar livre; com isto, o sistema apresentou taxas de erro 16.2% e 25.6% nos ambientes internos e externos, respectivamente, e após o uso de uma interface de correção que apresenta as possíveis palavras em ordem de probabilidade, estas taxas caíram para 1.2% e 2.2%.

Além da ferramenta Sphinx, temos o reconhecimento de voz da Google (utilizada pela API Speech Recognize do Android), que apresentava uma taxa de erro de 23% e vem caindo para 8% em apenas um ano. Segundo a empresa, a ferramenta não para de melhorar, se aprimorando constantemente através do uso, tendo a idéia de compreender o usuário e antecipar os próprios movimentos (TECMUNDO APUD GOOGLE I/O 2015, 2015).

DE PAULA (2013), descreve em seu trabalho o desenvolvimento de um aplicativo de coleta de dados georreferenciados através do reconhecimento de voz em Português Brasileiro ou Inglês, pela API Speech Recognize (disponibilizada pelo Android) visando, além disso, um layout simples e de fácil utilização.

Um fator negativo do reconhecimento de voz através da API Speech Recognize apontado no trabalho de DE PAULA (2013), foi a dependência de conexão com a Internet, já que a aplicação foi desenvolvida para o Android 2.2 (Froyo). Entretanto, a partir da versão 4.1 do Android (Jelly Bean), há o reconhecimento de voz off-line para digitação de textos, junto ao teclado nativo (voice typing).

Sphinx¹: estrutura flexível, modular e plugável para ajudar a promover inovações na pesquisa de sistemas de reconhecimento de voz (WALKER, 2004)

3

Aspectos Tecnológicos

Apresenta-se neste capítulo uma série de recursos tecnológicos que foram utilizados no desenvolvimento deste projeto.

3.1 Web Crawler

A World-Wide-Web é descentralizada, dinâmica e diversa, e encontrar informações nela pode ser um desafio. Este problema é chamado de problema da descoberta de recursos. O Web Crawler é uma ferramenta que resolve este problema, no contexto da World-Wide Web. Ele pode ser entendido como um "robô Web", que acessa um documento da Web por vez e opera, utilizando a infra-estrutura da página Web, para recuperar (adquirir) os dados ali presentes (PINKERTON, 1994).

Depois de recuperar um documento, o Web Crawler executa três ações: marca o documento como já recuperado, decifra quaisquer links externos e internos do conteúdo do documento e armazena as informações numa base de dados (PINKERTON, 1994).

Na ausência do acesso aos dados do Porteiro Web, foi construído um Web Crawler para recuperar os dados que foram usados no projeto e que serão usados no funcionamento da aplicação.

3.1.1 Python

Python é uma linguagem de programação interpretada, interativa e orientada a objetos. Ela fornece estruturas de dados de alto nível tais como lista e matrizes associativas, tipagem dinâmica, classes, exceções, gerenciamento automático de memória, etc. Python também tem uma sintaxe extraordinariamente simples e elegante (SANNER, 1999).

Novos módulos de extensão podem ser criados para estender a linguagem com código novo ou legado. Há um grande número de módulos de extensão que foram desenvolvidos e são distribuídos pelos membros da comunidade de usuários Python (SANNER, 1999).

Python foi utilizado neste projeto junto ao framework Scrapy (descrito na seção 3.1.1.1) para a construção do Web Crawler.

3.1.1.1 Scrapy

Scrapy é um framework de extração de informações web, utilizado para criar Web Crawlers, escrito em Python, com uma ampla gama de aplicações úteis, como mineração de dados e processamento de informação (WANG e GUO, 2012).

Lançado oficialmente em Agosto de 2008, Scrapy conta com 1500 seguidores no github e é uma das mais populares ferramentas de construção de crawlers e extração de informações web (PANTA, 2015).

3.2 Web Service

Web Services podem ser definidos como unidades modulares auto-suficientes de lógica de aplicação que oferece funcionalidade de negócio para outros aplicativos através da internet (SRIVASTAVA e KOEHLER, 2003).

Eles oferecem serviços para aplicações e podem ser vistos como objetos ou chamadas de funções através da internet, por meio de protocolos desenvolvidos para a web, podendo então, serem acessados de qualquer lugar do planeta Terra. Web Services, também são utilizados para integrar sistemas e aplicações distintas através da internet (AHO, SETHI, ULLMAN, 1986).

Neste trabalho, foi construído um Web Service para receber as requisições vindas do aplicativo Android, contendo as perguntas do usuário e encaminhá-las à camada de aplicação para serem processadas. Posteriormente, os resultados são disponibilizados na resposta do Web Service para as requisições feitas pelo aplicativo.

3.2.1 Web Service RESTful (REST)

Serviços Web RESTful são aplicações Web construídas sobre a arquitetura REST, onde arquitetura, dados e funcionalidades são consideradas recursos e estes recursos são acessados frequentemente através de links da web, denominados como URIs (Uniform Resource Identifiers) (HAMAD, SAAD, ABED, 2010).

O URI, em português, Identificador Uniforme de Recursos, é uma cadeia de caracteres compacta, utilizada para identificar ou denominar um recurso na Internet (WIKIPEDIA,2015).

Além disto, pode-se projetar Web services que especificam como os estados dos recursos são endereçados e transferidos, através do protocolo HTTP, para uma gama de clientes implementados em diferentes linguagens de programação (RODRIGUEZ, 2008).

Web Services RESTful são aplicações web construídas sobre a arquitetura REST, expondo recursos (dados e funcionalidades) através de URIs web, e possibilitando criar, recuperar, atualizar e deletar recursos através do protocolo HTTP. Nesta arquitetura, clientes e servidores trocam representações de recursos, utilizando interface e protocolo padronizados, proporcionando simplicidade, leveza e alta performance às aplicações REST. Além disto, resultados de comparação de performance entre o RESTful e seu concorrente SOAP mostram claramente a superioridade do RESTful sobre o SOAP, conforme mostra a Figura 3 (HAMAD, SAAD, ABED, 2010).

Number of array elements I	Message Size (byte)				Time (Milliseconds)			
	SOAP/HTTP		REST (HTTP)		SOAP/HTTP		REST (HTTP)	
	String Concatenation	Float Numbers Addition	String Concatenation	Float Numbers Addition	String Concatenation	Float Numbers Addition	String Concatenation	Float Numbers Addition
2	351	357	39	32	781	781	359	359
3	371	383	48	36	828	781	344	407
4	395	409	63	35	828	922	359	375
5	418	435	76	39	969	1016	360	359
6	443	461	93	43	875	953	359	359
7	465	487	104	47	875	875	469	360
8	493	513	127	51	984	875	437	344

Figura 3 - Comparativo entre Web Service RESTful e SOAP

Pode-se notar a diferença de tamanho da mensagem (em bytes) e de tempo gasto no processamento (em milissegundos), entre o SOAP e o REST, sendo o REST muito mais econômico e rápido do que o SOAP.

3.2.2 Arquivo JSON

JSON (JavaScript Object Notation) é uma linguagem [de marcação] que foi projetada para facilitar a análise, a troca e o uso de dados entre computadores, sendo também a mais adequada para aplicações JavaScript, proporcionando um ganho de desempenho significativo se comparado à linguagem [de marcação] XML (eXtensible Markup Language), que requer bibliotecas adicionais para recuperar dados de objetos DOM (Document Object Model) (NURSEITOV, 2009).

Além do que foi citado, vale frisar também que a sintaxe do JSON é legível para humanos e utiliza contexto para evitar ambiguidades, assim como as linguagens de programação fazem. Os resultados indicam que JSON é mais rápido e utiliza menos recursos do que o seu concorrente XML (NURSEITOV, 2009). Estima-se que a leitura de um arquivo JSON é cerca de cem vezes mais rápida do que a leitura de um arquivo XML (JSON, 2015).

Neste projeto, JSON é utilizado para organizar a resposta do Web Service à requisição feita pelo aplicativo android com a pergunta do usuário. Na Figura 4 é mostrado um trecho da resposta do Web Service em formato JSON.

```
{
  "eventos": [{ "titulo": "REDES",
    "periodo_curso": "5º COMP",
    "professor": "PROF. FLÁVIO BARBIERI",
    "descricao": "5º COMP PROF. FLÁVIO BARBIERI",
    "horario_inicial": "Segunda-feira, 15 de Junho às 14h00",
    "horario_final": "Segunda-feira, 15 de Junho às 16h00",
    "sala": "PRÉDIO B - B-302",
    "similaridade": 10,
    "data_inicial": "Jun 15, 2015 2:00:00 PM"}]
}
```

Figura 4 - Trecho em JSON com objetos que constituem a lista de eventos

3.3 Android

Android é um software para dispositivos móveis que inclui um sistema operacional, middleware e aplicações-chave (DEVELOPERS, 2011).

Oriundo de uma startup de 22 meses de idade, o Android foi comprado discretamente pela Google em 2005 (BLOOMBERG BUSINESS, 2005).

O sistema operacional vem sendo, então, desenvolvido intensamente nos últimos 15 anos, começando pelos telefones pretos e brancos até os recentes smartphones ou mini computadores, e é um dos sistemas móveis mais utilizados atualmente e após o seu lançamento, muitas atualizações ocorreram. Além disso, o Android fornece APIs necessárias para o desenvolvimento de aplicações para a sua plataforma, utilizando a linguagem de programação Java (JADHAV e PATIL, 2012).

3.3.1 Speech Recognizer

As APIs disponibilizadas pelo Android possuem diversas funcionalidades que permitem o uso dos serviços do sistema, tendo como destaque sua otimização. Através delas, pode-se criar todas as interfaces com o usuário, permitindo a criação de telas, acesso a arquivos, criptografia de dados, entre outras coisas. Uma destas APIs é a Speech Recognizer, para reconhecimento de voz em diversas linguagens, independente de locutor, para falas contínuas ou isoladas (DE PAULA, 2013).

4 Fundamentação Teórica e Metodologia

Este capítulo detalha o desenvolvimento da aplicação, explicando cada uma das partes que compõem o projeto.

4.1 Arquitetura do Sistema

Na Figura 5 é mostrada a arquitetura elaborada para a aplicação desenvolvida.

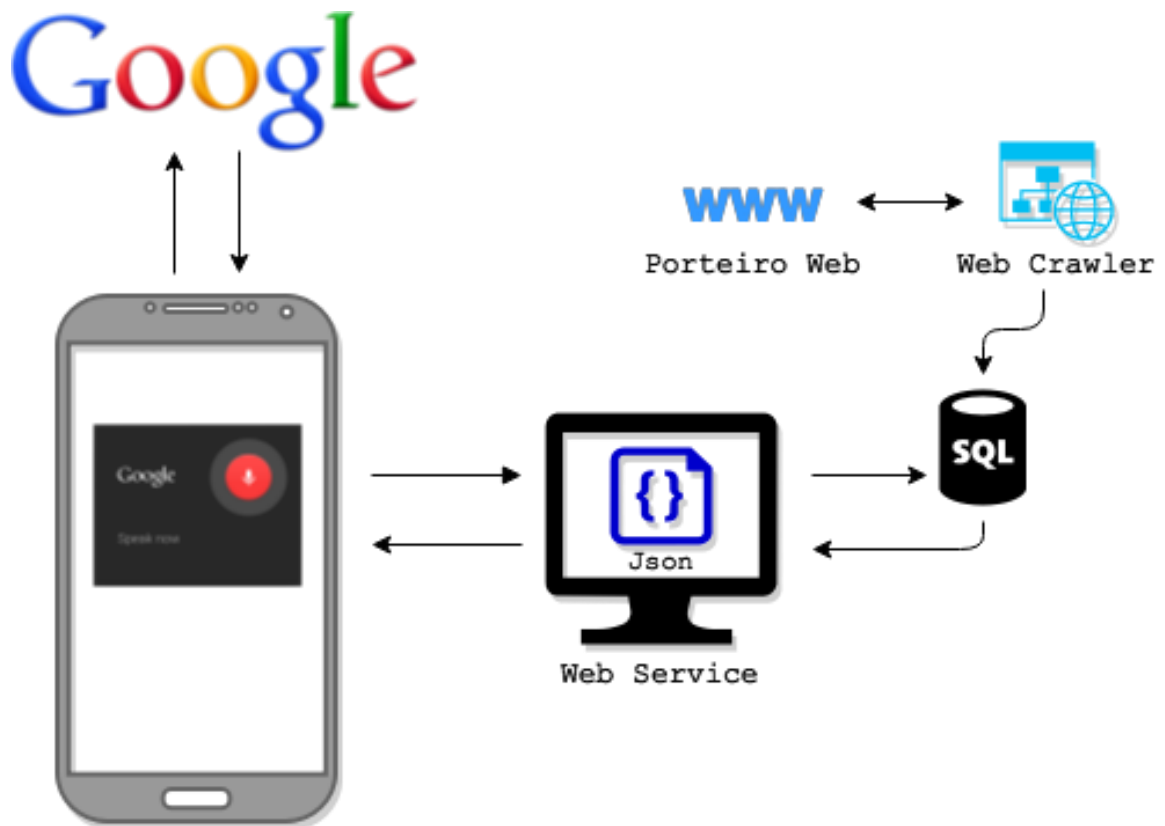


Figura 5 - Arquitetura utilizada no projeto

Para possibilitar o processamento das informações contidas no sistema Porteiro Web é necessário que se tenha todas as suas informações persistidas em

um banco de dados, e para atender a isto, foi necessário desenvolver um Web Crawler que captura e persiste as informações do sistema Porteiro Web, pois o acesso aos dados do sistema não é disponibilizado publicamente.

O aplicativo Android, por sua vez, recebe a pergunta realizada pelo usuário por entrada de voz (tendo a fala do usuário enviada aos servidores da Google para transcrição da mesma para texto) ou por entrada de texto e a mesma é enviada dentro de uma URL para o Web Service, onde é tratada pela aplicação desenvolvida, extraindo da pergunta somente os dados necessários para a busca e aplicando o algoritmo de similaridade local (entre os dados extraídos da pergunta do usuário e os dados do Porteiro Web, que estão no banco de dados).

Após o processamento descrito, o resultado é recuperado em formato JSON na resposta do Webservice e exibido de forma agradável na tela do dispositivo para o usuário.

4.2 Estudo do Porteiro Web

O Porteiro Web é um sistema da UNIFAL-MG, destinado ao gerenciamento de eventos e reserva de salas.

Seu conteúdo é constantemente modificado e incrementado por funcionários da instituição e está armazenado em um banco de dados; porém, não é disponibilizado publicamente.

A disponibilização pode ser feita através de leitura direta no banco de dados ou através de arquivos *dumps* (arquivo que contém as estruturas das tabelas e/ou os dados contidos em um banco de dados), como é feito por diversas bibliotecas na Internet, incluindo a Wikipedia, uma das maiores enciclopédias virtuais, possuindo bases de dados gigantescas disponibilizadas publicamente em dumps, conforme mostra a Figura 6. Com este tipo de arquivo, é possível importar todos os dados para um banco local.

enwiki dump progress on 20150602

This is the Wikimedia dump service. Please read the [copyrights](#) information.

See [all databases list](#).

[Last dumped on 2015-05-15](#)

Dump complete

Verify downloaded files against the [MD5 checksums](#) to check for corrupted files.

```
2015-06-16 21:27:35 done Articles, templates, media/file descriptions, and primary meta-pages, in multiple bz2
enwiki-20150602-pages-articles-multistream.xml.bz2 11.9 GB
enwiki-20150602-pages-articles-multistream-index.txt.bz2 162.8 MB

2015-06-16 18:10:18 done All pages with complete edit history (.7z)
enwiki-20150602-pages-meta-history1.xml-p000000010p000002884.7z 224.7 MB
enwiki-20150602-pages-meta-history1.xml-p000002885p000005134.7z 211.7 MB
enwiki-20150602-pages-meta-history1.xml-p000005137p000007399.7z 221.2 MB
enwiki-20150602-pages-meta-history1.xml-p000007400p000009697.7z 230.5 MB
enwiki-20150602-pages-meta-history1.xml-p000009700p000010001.7z 30.0 MB
enwiki-20150602-pages-meta-history2.xml-p000010001p000012463.7z 219.7 MB
enwiki-20150602-pages-meta-history2.xml-p000012464p000014579.7z 218.7 MB
enwiki-20150602-pages-meta-history2.xml-p000014580p000016142.7z 172.4 MB
enwiki-20150602-pages-meta-history2.xml-p000016143p000017909.7z 157.8 MB
enwiki-20150602-pages-meta-history2.xml-p000017910p000019514.7z 154.8 MB
enwiki-20150602-pages-meta-history2.xml-p000019515p000021211.7z 164.8 MB
enwiki-20150602-pages-meta-history2.xml-p000021212p000023000.7z 172.1 MB
enwiki-20150602-pages-meta-history2.xml-p000023001p000025001.7z 200.6 MB
enwiki-20150602-pages-meta-history3.xml-p000025001p000026989.7z 206.1 MB
enwiki-20150602-pages-meta-history3.xml-p000026990p000029782.7z 236.6 MB
```

Figura 6 - DUMP disponibilizado pela Wikipedia

Devido a este fato, foi necessário o desenvolvimento de um Web Crawler para que fosse possível adquirir e armazenar as informações contidas no sistema.

Além do problema da disponibilidade dos dados descrito acima, observou-se durante o desenvolvimento deste trabalho, uma visível falta de padronização no conteúdo do sistema, através de abreviaturas e de truncamento de palavras ou de nomes, impedindo muitas vezes que fosse encontrado a resposta esperada pelo usuário, tornando necessário a utilização de um algoritmo que calcula o grau de similaridade entre palavras.

Para que se tenha uma busca no mínimo satisfatória, com os resultados esperados, todos os dados referentes a cada evento no sistema, como nome e descrição do evento (aula, palestra, etc) e nome do responsável pela execução do evento (professor, ministrante, etc), devem ser preenchidos por completo e sem abreviaturas, pois o usuário dificilmente fará uma busca utilizando abreviações, e

caso ele busque utilizando o nome completo de uma disciplina ou de um professor, por exemplo, se os mesmos se apresentarem truncados ou abreviados no banco de dados, haverá um grande risco desta informação não ser encontrada pela busca.

4.3 Desenvolvimento do Web Crawler

O Web Crawler presente neste trabalho, é o responsável por capturar as informações contidas no sistema Porteiro Web e foi desenvolvido na linguagem de programação Python, com o auxílio do Framework Scrapy, devido a abundância do mesmo em recursos disponíveis para o desenvolvimento de Web Crawlers.

Para elaborar um Web Crawler é necessário, primeiramente, definir quais elementos das páginas web que se deseja obter e no caso deste projeto, os dados desejados são as informações de todos os eventos dos dois Campi da UNIFAL-MG, Sede e Santa Clara, disponíveis no Porteiro Web.

Através do Scrapy, é possível percorrer todos os links correspondentes a todos os eventos de cada página do sistema, onde cada página corresponde unicamente a um dia, de uma sala, bem como depende do Campus, sala e data desejada. Portanto, ao pesquisar alguma informação sobre determinado evento no Porteiro Web, é necessário saber em qual Campus, sala, dia, mês e ano o mesmo ocorrerá.

Ao analisar as URLs do sistema, encontrou-se um padrão, descrito na Figura 7, onde as URLs são sempre iguais, mudando somente informações específicas como Campus, sala, dia, mês e ano.

URL: <https://www.unifal-mg.edu.br/app/graduacao/porteiroweb/day.php?year=2015&month=06&day=18&area=3>

Data: 18/06/2015

Campus: Sede

Prédio: PCA (área 3)

URL: <https://www.unifal-mg.edu.br/app/graduacao/porteirowebstaclara/day.php?year=2015&month=06&day=19&area=32>

Data: 19/06/2015

Campus: Santa Clara

Prédio: B (área 32)

Figura 7 - Padrão das URLs do sistema Porteiro Web

Para gerar a lista de URLs que alimentará o Scrapy, foi determinado um intervalo constante de 3 meses, a partir da data atual, no qual o sistema irá gerar todas as URLs correspondentes à todos os dias dos 3 meses, abrangendo todas as salas e prédios dos Campus Sede e Santa Clara.

O Scrapy percorrerá essa lista, acessando cada URL (onde cada uma abrirá uma página correspondente) e extraíndo todos os links de cada página. Estes links extraídos, correspondem cada um a um evento do sistema, e o Scrapy também os acessará, chegando, finalmente à página onde se encontra as informações específicas e detalhadas do evento acessado. Esta página é mostrada na Figura 8.

Porteiro Web - Agendamento de Salas
Universidade Federal de Alfenas - Unidade II

1 ▾ Jul ▾ 2015 ▾ Ver Agenda Pesquisa:

ALGORÍTMOS

Descrição: 1º COMP - PROF. PAULO BRESSAN/LEANDRA
Agendado para:
Sala: PRÉDIO B - B-202
Horário Inicial: 10:00-11:00, quarta 01 julho 2015
Duração: 2 horário(s)
Horário Final: 11:00-12:00, quarta 01 julho 2015
Tipo de Evento: Aula Graduação
Agendado por: SERGILENE ROCHA - sergilene.rocha@unifal-mg.edu.br - 1078
Última Atualização: 13:52:11 - segunda 12 janeiro 2015
Repetir Agendamento: Semanalmente
Repetir no Dia: quarta
Repetir iniciando na Data: quarta 25 fevereiro 2015
Repetir até a Data: quinta 09 julho 2015

Figura 8 - Página com a descrição completa de um evento no Porteiro Web

Para que fosse possível capturar todas estas informações de cada evento, foi necessário analisar o código fonte da página para saber qual elemento extrair. O código fonte da Figura 9, corresponde ao evento mostrado anteriormente na Figura 8.

```
<body bgcolor="#ffffed" text="black" link="#5B69A6" vlink="#5B69A6" alink="red">
  <div id="header">...</div>
  <table width="100%" border="0" cellpadding="0" cellspacing="0">...</table>
  <h3>ALGORITMOS</h3>
  <table border="0">
    <tbody>
      <tr>
        <td align="right">...</td>
        <td>1º COMP - PROF. PAULO BRESSAN/LEANDRA</td>
      </tr>
      <tr>
        <td align="right">...</td>
        <td>...</td>
      </tr>
      <tr>
        <td align="right">...</td>
        <td>PRÉDIO B - B-202</td>
      </tr>
      <tr>
        <td align="right">...</td>
        <td>10:00-11:00, quarta 01 julho 2015</td>
      </tr>
      <tr>
        <td align="right">...</td>
        <td>2 horário(s)</td>
      </tr>
      <tr>
        <td align="right">...</td>
        <td>11:00-12:00, quarta 01 julho 2015</td>
      </tr>
      <tr>
        ...
      </tr>
    </tbody>
  </table>
</body>
```

Figura 9 - Código fonte da página de descrição de um evento no Porteiro Web

No Scrapy conseguiu-se percorrer cada tag HTML presente no código fonte, e como pode-se observar, as informações desejadas encontram-se de forma padronizada dentro de tags específicas. Por exemplo, para capturar o título do evento, é preciso percorrer as tags `<table>` e `<h3>` e para capturar o restante dos conteúdos do evento, é necessário percorrer as tags `<table>`, `<tr>` e `<td>`.

Após todo o processo de geração e acessos das URLs e extração das informações, o conteúdo será armazenado no banco de dados do sistema. Vale frisar também que o banco é atualizado diariamente, de forma automática, através do Web Crawler, que será invocado uma vez ao dia durante a madrugada (período de tráfego menos intenso do sistema), por um Script armazenado no servidor da aplicação.

4.4 Elaboração das Perguntas

As possíveis perguntas a serem realizadas pelos usuários foram baseadas nas informações sobre eventos que são disponibilizadas pelo sistema Porteiro Web; desta forma, todas as possíveis perguntas giram em torno de informações como locais, horários, professores e cursos.

Durante a elaboração das possíveis perguntas, também levou-se em conta vícios de linguagem, erros de português e diversas outras possíveis variantes. Um esboço utilizado para elaborar as perguntas é mostrado abaixo na Figura 10:

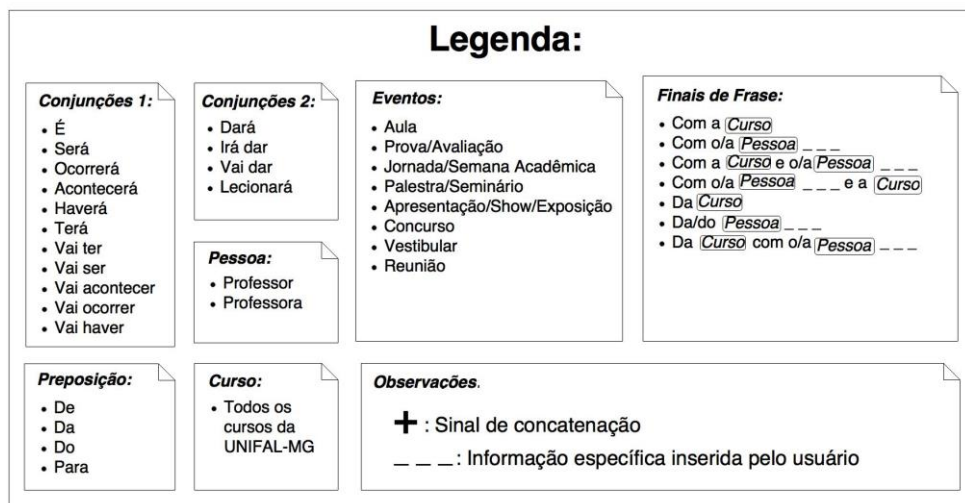
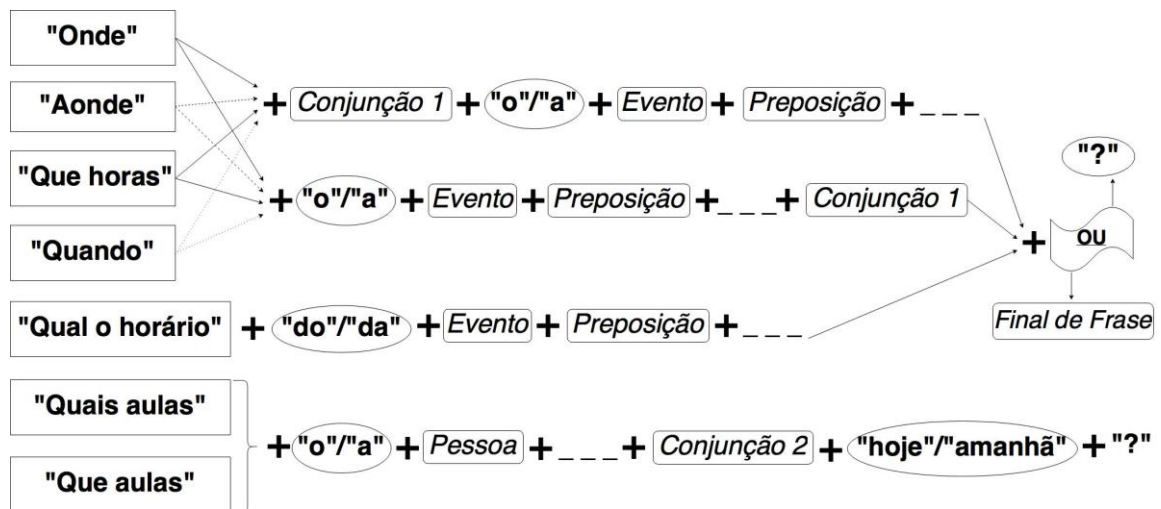


Figura 10 - Esboço utilizado na elaboração das possíveis perguntas

4.5 Extraindo dados das perguntas

Para efetuar a busca das informações desejadas, é necessário que se tenha em mãos alguma informação pertinente ao que se deseja buscar, para que seja possível elaborar uma *query* (comandos em um SGBD, como busca, inserção, etc) de busca consistente. Neste trabalho, presume-se que a informação esteja contida na pergunta do usuário, necessitando de várias manipulações para extrair estes dados.

Para tornar possível a extração, foi criado um padrão de entrada para as perguntas, ou seja, é necessário a identificação de ao menos uma informação (palavra-chave), dentre tipo de evento, nome do responsável (professor, ministrante, etc) ou nome do curso, antes da descrição da mesma.

Por exemplo, na pergunta “Onde será a aula de Algoritmos com o professor Bressan?”, tem-se como identificadores (ou palavras-chave), “aula” e “professor”; e como informação, “Algoritmos” e “Bressan”. Assim, a aplicação será capaz de retornar resultados concisos, pois foi possível determinar que a busca se refere à um evento do tipo “aula” com um determinado professor. Contudo, caso esta pergunta fosse elaborada desta forma: “Onde será Algoritmos com o Bressan”, a mesma seria ignorada, devido à falta de identificadores, para ao menos um dos três parâmetros da pesquisa, como tipo de evento ou pessoa responsável ou até mesmo curso pertinente.

Uma coleção de palavras-chave foi criada abrangendo as possibilidades existentes no Porteiro Web, portanto, basta a pergunta efetuada ser adequada ao padrão descrito, que se a informação estiver presente no banco de dados, será retornada ao usuário.

4.5.1 Stop Words

Stop-words são palavras comumente utilizadas e são removidas por possuírem pouco significado, incluindo artigos, conjunções, preposições, sinais de pontuação, números, caracteres não alfabéticos, etc. (KOBBERSTEIN, 2006)

Palavras como "e", "ou", "o", e "uma", possuem pouco significado e aparecem com relativa frequência em todos os documentos e, assim, não fornecem muita informação para distinguir um documento de outro. (KOBBERSTEIN, 2006)

Comumente, são julgadas como *stop words*, palavras classificadas como preposições, artigos e verbos de ligação, variando conforme o objetivo desejado, pois estas palavras não possuem especificidade, isto é, podem ser utilizadas em todo e qualquer contexto e nada irão inferir sobre algum assunto ou tema em específico, servindo apenas para a compreensão da língua, seja ela falada ou escrita.

Neste trabalho, a construção da lista de *stop words* engloba somente o idioma Português Brasileiro e foi baseada parcialmente na lista disponível em (SNOWBALL, 2015), sendo consideradas, além das preposições, artigos e verbos de ligação, palavras específicas do contexto da aplicação, selecionadas com base nas possíveis perguntas dos usuários para o aplicativo.

Exemplo:

Pergunta do usuário: “Onde será a aula de Algoritmos com o professor Bressan?”

Pergunta sem stop-words: “aula Algoritmos professor Bressan”

Após a exclusão das *stop words* da pergunta feita pelo usuário, tem-se em mãos apenas o necessário para prosseguir com o restante do processamento.

4.5.2 Capturando dados para a busca

Logo após remover as *stop words* da pergunta do usuário, o próximo passo é identificar o conteúdo específico de cada elemento (evento, pessoa e curso) para realizar a busca.

O algoritmo irá extrair os dados resumidamente da seguinte forma:

- Recebe a string da pergunta sem as *stop-words*;
- Quebra a string recebida no espaçamento entre as palavras, armazenando-as em um vetor;

- Percorre este vetor de forma a encontrar palavras-chaves pré-definidas (aula, reunião, show, professor, curso, etc);
- Ao encontrar uma palavra-chave, ele a identifica como sendo um evento, nome de professor, nome de curso, ou horário, determinando o tipo da informação que sucede a palavra-chave na string;
- Após saber o tipo, ele irá capturar esta informação que será utilizada ao longo da aplicação (Exemplo: da frase “aula Algoritmos”, a palavra-chave seria “aula” e o conteúdo seria “Algoritmos”);
- Depois, ele continua percorrendo o vetor de strings até encontrar outra palavra-chave e capturar a informação ou até o fim da string;

Considerando a possível pergunta de um usuário: “aula Algoritmos professor Bressan”, seguindo o algoritmo descrito, tem-se:

- Vetor de strings: {aula, Algoritmos, professor, Bressan};
- Palavras-chave, especificando o tipo de evento: “aula” e “professor”;
- Conteúdo: “Algoritmos” e “Bressan”;

Contudo, analisando o esboço da Figura 10, considerando o processo de remoção das *stop words* e de captura dos dados para busca, concluiu-se que seria possível inserir perguntas na aplicação de várias outras formas, apenas seguindo o padrão mostrado na Figura 11:

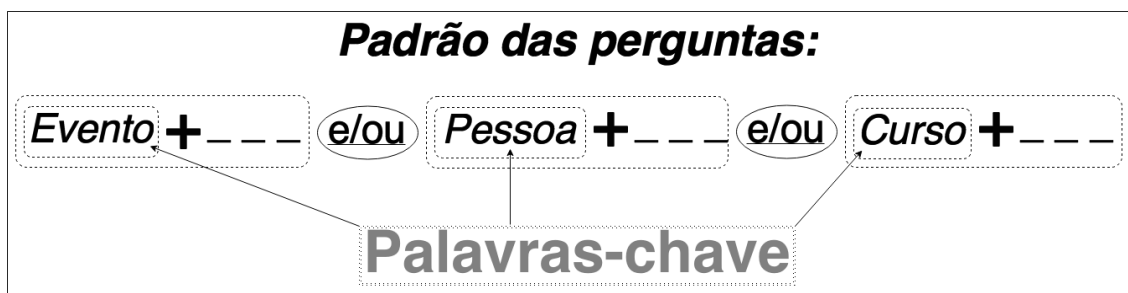


Figura 11 - Ilustração da padronização

4.6 Obtendo os resultados a partir do banco de dados

Após o processo de extração dos dados para a busca, tem-se ao menos uma, de três variáveis (evento, pessoa e curso), preenchida com seu respectivo conteúdo.

Devido a falta de padronização e à grande quantidade de abreviaturas e truncamentos existentes na base de dados extraída do sistema, o uso do conteúdo das variáveis diretamente inseridas na *query* tornou-se inviável, pois mesmo utilizando um recurso para aumentar a abrangência da busca (operador SQL “LIKE”), muitos resultados esperados não eram obtidos, como se a informação não existisse. O operador “LIKE” e as complicações descritas serão explicadas nas subseções 4.5.1 e na seção 4.6, respectivamente.

4.6.1 MySQL - Operações utilizando o operador “LIKE”

O operador “LIKE” pertence à linguagem estruturada de consultas conhecida como SQL (*Structured Query Language*) e é muito comum e utilizada durante as operações relacionadas às consultas em banco de dados, principalmente para consultas envolvendo texto.

Pode ser utilizado juntamente com o símbolo “%” para auxiliar na busca pelo banco, podendo ser inseridos antes e/ou depois da string desejada, bem como em seu interior, fazendo com que, ao longo do conteúdo do banco, todo e qualquer caractere que esteja localizado no lugar do símbolo “%” seja considerado como resultado da busca.

Na Figura 12, é demonstrado o uso do operador junto ao símbolo auxiliar “%”.

```
mysql> SELECT titulo, periodo_curso, professor, sala FROM eventos WHERE titulo LIKE 'física' GROUP BY titulo;
Empty set (0.00 sec)

mysql> SELECT titulo, periodo_curso, professor, sala FROM eventos WHERE titulo LIKE 'física%' GROUP BY titulo;
+-----+-----+-----+-----+
| titulo          | periodo_curso          | professor          | sala          |
+-----+-----+-----+-----+
| FÍSICA AMB     | CARÁTER ESPECIAL FÍSICA | PROF. CÉLIO        | PAVILHÃO CENTRAL DE AULAS - PCA-302 |
| FÍSICA I       | 2º QUÍM BACH           | PROF. IHOSVANY CAMPS | PAVILHÃO CENTRAL DE AULAS - PCA-204 |
| FÍSICA II      | 3º FÍS                 | PROF. CRISTIANA SCHMIDT | PAVILHÃO CENTRAL DE AULAS - PCA-204 |
| FÍSICA III     | 4º QUÍM BACH           | PROF. MIRTA MIR CARABALLO | PAVILHÃO V - V-311 |
| FÍSICA IV      | 5º FÍS                 | PROF. PERSON NEVES   | PAVILHÃO CENTRAL DE AULAS - PCA-205 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT titulo, periodo_curso, professor, sala FROM eventos WHERE titulo LIKE '%física' GROUP BY titulo;
+-----+-----+-----+-----+
| titulo          | periodo_curso          | professor          | sala          |
+-----+-----+-----+-----+
| BIOFÍSICA      | 2º CB LIC              | PROF. CONCURSO EDITAL 089/2014 | PAVILHÃO 0 - 0 326 |
| COMP FÍSICA    | 1º FÍS                 | PROF. SAMUEL BUENO   | PAVILHÃO CENTRAL DE AULAS - PCA-203 |
| UNATI - ATIVIDADE FÍSICA | UNATI ATIVIDADE FÍSICA |                          | PAVILHÃO L - L106 |
+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> SELECT titulo, periodo_curso, professor, sala FROM eventos WHERE titulo LIKE '%física%' GROUP BY titulo;
+-----+-----+-----+-----+
| titulo          | periodo_curso          | professor          | sala          |
+-----+-----+-----+-----+
| BIOFÍSICA      | 2º CB LIC              | PROF. CONCURSO EDITAL 089/2014 | PAVILHÃO 0 - 0 326 |
| COMP FÍSICA    | 1º FÍS                 | PROF. SAMUEL BUENO   | PAVILHÃO CENTRAL DE AULAS - PCA-203 |
| FÍSICA AMB     | CARÁTER ESPECIAL FÍSICA | PROF. CÉLIO        | PAVILHÃO CENTRAL DE AULAS - PCA-302 |
| FÍSICA I       | 2º QUÍM BACH           | PROF. IHOSVANY CAMPS | PAVILHÃO CENTRAL DE AULAS - PCA-204 |
| FÍSICA II      | 3º FÍS                 | PROF. CRISTIANA SCHMIDT | PAVILHÃO CENTRAL DE AULAS - PCA-204 |
| FÍSICA III     | 4º QUÍM BACH           | PROF. MIRTA MIR CARABALLO | PAVILHÃO V - V-311 |
| FÍSICA IV      | 5º FÍS                 | PROF. PERSON NEVES   | PAVILHÃO CENTRAL DE AULAS - PCA-205 |
| MÉT FÍSICA II  | 5º FÍS                 | PROF. ARTUR JUSTINIANO | PAVILHÃO CENTRAL DE AULAS - PCA-302 |
| UNATI - ATIVIDADE FÍSICA | UNATI ATIVIDADE FÍSICA |                          | PAVILHÃO L - L106 |
+-----+-----+-----+-----+
9 rows in set (0.02 sec)
```

Figura 12 - Exemplo de query SQL utilizando o operador LIKE e o símbolo %

Na primeira *query*, não foi utilizado o símbolo auxiliar “%” e nenhum resultado foi exibido, pois não há nenhuma string exatamente igual à busca. Já na segunda e terceira *query*, foi utilizado o símbolo “%” no final e no início da string, respectivamente, obtendo como resultado, as strings iniciadas com “física” e as strings terminadas com “física”.

Na última *query*, onde aparece a maior quantidade de resultados, é utilizado o símbolo “%” no início e no final da string, e os resultados exibidos são os que possuem “física” tanto em seu interior, como em seu início e fim.

Desta forma, com o símbolo “%” inserido no início e no fim da string desejada, amplia-se a possibilidade da busca, abrangendo e trazendo mais resultados, e portanto, esta foi a escolha de utilização do operador “LIKE” junto ao símbolo “%”, no início deste projeto, para realizar as buscas.

No entanto, em virtude da falta de padronização e à grande quantidade de abreviaturas e truncamentos presentes na base de dados, a utilização do operador descrito precisou ser descartada, como é descrito na seção 4.6.

4.7 Complicações

O uso do operador “LIKE” junto ao símbolo “%” na consulta composta pelo conteúdo direto das informações extraídas da pergunta do usuário, seria uma solução viável e abrangente para a nossa aplicação, porém, o grande uso de abreviaturas e truncamentos de nomes de eventos, disciplinas, nomes de professores e etc, no conteúdo do sistema, impedia que muitos resultados esperados fossem exibidos tornando, então, inviável o seu uso.

Os dados extraídos e persistidos pelo Web Crawler no banco de dados, são capturados diretamente do sistema Porteiro Web, portanto, da mesma forma que a informação foi inserida no sistema, ela irá para a base de dados; ou seja, truncadas, abreviadas ou contendo apenas siglas, em maioria incomuns e não utilizadas no cotidiano, como “TCC” (abreviatura da disciplina “Trabalho de Conclusão de Curso”), por exemplo. Este fato é totalmente prejudicial à busca, pois a pergunta a ser feita pelo usuário, raramente terá forma abreviada incomum ou somente siglas incomuns, como é mostrado em destaque na Figura 13:

Horários:	Sala: V-202 (Cap.: 60)	Sala: V-203 (Cap.: 50)	Sala: V-204 (Cap.: 50)	Sala: V-205 (Cap.: 60)	Sala: V-301 (Cap.: 70)	Sala: V-302 (Cap.: 70)	Sala: V-303 (Cap.: 50)	Sala: V-304 (Cap.: 50)	Sala: V-305 (Cap.: 50)
07:00-08:00	+	QUÍM GERAL	FISIO	Q. INORG I	PROCESSO SELETIVO - EDITAL N° 054/2015 -	+	+	+	+
08:00-09:00	MICRO	"	"	"	"	+	BIM MEDICINA	BIM MEDICINA	+
09:00-10:00	HISTO	EVOL PLANTAS	+	BIOQUÍM	"	IMUNO I	"	"	BIO CEL
10:00-11:00	"	"	TOXICO	"	"	"	"	"	"
11:00-12:00	+	+	+	+	"	+	+	+	+
12:00-13:00	+	+	+	+	"	PROCESSO SELETIVO - EDITAL N° 054/2015	+	+	+
13:00-14:00	GENÉTICA	TEC ALIM	MICRO	EDUC NUT (1)	TOLFT	"	BIO CEL	INT QUÍM	ANATO

Figura 13 - Tela exibindo eventos e mostrando as recorrentes abreviaturas

Considerando a possível pergunta de um usuário: “Onde será a aula de Tecnologia de Alimentos?” como exemplo, foi feita uma pesquisa no banco de dados (Figura 14), através de uma *query*, que busca o evento citado:

```
mysql> SELECT * FROM eventos WHERE titulo LIKE '%Tecnologia de Alimentos%';  
Empty set (0.09 sec)
```

Figura 14 - Query SQL com resultado vazio devido ao problema das abreviaturas

Como pode-se observar, nenhum resultado é retornado, pois no banco o registro do evento que corresponderia à pergunta do usuário, foi persistido como “TEC ALIM”.

A melhor maneira de solucionar este problema é padronizar todo o conteúdo do sistema; de nome de eventos e de professores a nomes de cursos, utilizando a forma completa de escrita, sem abreviações, truncamentos ou utilizando apenas siglas.

Para realizar estas modificações, seria necessário um tempo considerável, indo além do disponível para o desenvolvimento deste trabalho, o que tornou a opção inviável. A alternativa adotada, foi a implementação e a utilização de um algoritmo de similaridade entre palavras, que será detalhado na subseção 4.6.1.

4.7.1 Alinhamento de Strings

O alinhamento de strings pode ser considerado basicamente sob dois aspectos possíveis: Alinhamento Global e Alinhamento Local. As informações apresentadas nessa seção são baseadas em (GUSFIELD,1997).

“O Alinhamento Global de duas strings S_1 e S_2 é obtido através da inserção de espaços (ou traços), em diferentes pontos de S_1 e S_2 , e então posicionando-se ambas as strings resultantes uma sobre a outra de modo que todo caractere ou espaço em uma string esteja alinhado a um único caractere ou espaço na outra”.

“O Alinhamento Local de duas strings S_1 e S_2 consiste em encontrar substrings α e β de S_1 e S_2 respectivamente cujo alinhamento seja máximo dentre todas as possíveis substrings de S_1 e S_2 ”.

O termo global enfatiza, portanto, o fato de que o alinhamento envolve na totalidade ambas as strings, enquanto que o termo local deixa claro o fato de que se procura alinhar substrings. Assim, as seções 4.6.1.1 e 4.6.1.2 exibem exemplos de

Alinhamento Global e Local, e embasa a escolha do Alinhamento Local para o desenvolvimento do presente trabalho.

4.7.1.1 Alinhamento Global

No contexto de Alinhamento Global, uma técnica clássica na área de programação dinâmica¹ é o algoritmo de distância de edição (também conhecido como distância de Levenshtein). Esse algoritmo computa o custo de se transformar uma string S_1 em outra S_2 . Considere como exemplo ilustrativo do alinhamento global então encontrar o alinhamento máximo (que resulte no menor custo) para duas strings: $S_1 = \text{writers}$, e $S_2 = \text{vintner}$. O resultado desta operação é obtido computando-se a tabela da distância de edição entre as duas strings, mostrada na Figura 15, e explicada em sequência.

$D(i,j)$			w	r	i	t	e	r	s
		0	1	2	3	4	5	6	7
	0	0	← 1	← 2	← 3	← 4	← 5	← 6	← 7
v	1	↑ 1	↖ 1	↖ ← 2	↖ ← 3	↖ ← 4	↖ ← 5	↖ ← 6	↖ ← 7
i	2	↑ 2	↖ ↑ 2	↖ 2	↖ 2	← 3	← 4	← 5	← 6
n	3	↑ 3	↖ ↑ 3	↖ ↑ 3	↖ ↑ 3	↖ 3	↖ ← 4	↖ ← 5	↖ ← 6
t	4	↑ 4	↖ ↑ 4	↖ ↑ 4	↖ ↑ 4	↖ 3	↖ ← 4	↖ ← 5	↖ ← 6
n	5	↑ 5	↖ ↑ 5	↖ ↑ 5	↖ ↑ 5	↑ 4	↖ 4	↖ ← 5	↖ ← 6
e	6	↑ 6	↖ ↑ 6	↖ ↑ 6	↑ 6	↑ 5	↖ 4	↖ ← 5	↖ ← 6
r	7	↑ 7	↖ ↑ 7	↖ 6	↖ ← ↑ 7	↑ 6	↑ 5	↖ 4	← 5

Figura 15 - Comparação das strings writers e vintner utilizando alinhamento global

Programação Dinâmica¹: Método de construção de algoritmos, cuja solução pode ser computada utilizando resultados já anteriormente calculados e memorizados, evitando retrabalho (WIKIPEDIA, 2015).

A primeira observação pertinente à Figura 15 é que conforme pode-se observar, a complexidade do algoritmo em questão é de $O(nm)$, onde n e m são os respectivos tamanhos de ambas as strings em questão. O preenchimento cobre as seguintes operações nos caracteres de ambas as strings: I - Insertion (inserção), D - Deletion (deleção), R - Replacement (substituição), M - Match (casamento). Uma seta na horizontal, indo da célula (i, j) para a célula $(i, j-1)$, representa uma operação de inserção (I) do caractere $S2(j)$ em $S1$; uma seta vertical, indo da célula (i, j) para a célula $(i-1, j)$, representa uma operação de deleção (D) do caractere $S1(i)$ de $S1$; e uma seta na diagonal, indo da célula (i, j) para a célula $(i-1, j-1)$ representa uma operação de casamento (M) se $S1(i) = S2(j)$, ou uma substituição (R) se $S1(i) \neq S2(j)$.

O valor final na tabela (contido na célula de maior valor i e j), indica o custo de alinhamento entre as strings $S1$ e $S2$, que no exemplo em questão vale 5. Isso significa que com 5 operações (I, D ou R), é possível se transformar $S1$ em $S2$. Ao se seguir as setas do maior i e j até a posição $i = 0$ e $j = 0$, observa-se a possibilidade de três caminhos diferentes. Essas possibilidades indicam justamente a existência de três possíveis alinhamentos diferentes, e que resultariam no mesmo custo ótimo (igual a 5), mostrados nas tabelas 1, 2 e 3. As colunas destacadas são exatamente aquelas que resultaram em penalização (caso alguma das operações I, D ou R tenham sido executadas).

Tabela 1 - Alinhamento Possível I

w	r	i	t	-	e	r	s
v	i	n	t	n	e	r	-

Tabela 2 - Alinhamento Possível II

w	r	i	-	t	-	e	r	s
v	-	i	n	t	n	e	r	-

Tabela 3 - Alinhamento Possível III

w	r	i	-	t	-	e	r	s
-	v	i	n	t	n	e	r	-

4.7.1.2 Alinhamento Local

O Alinhamento Local, conforme já citado, busca então encontrar substrings de máxima semelhança. Assim, ao contrário da distância de edição, que mede a diferença entre strings, o alinhamento local trabalha com o conceito oposto, buscando maximizar a similaridade. É uma implementação semelhante à de distância de edição já apresentada, inclusive contendo a mesma complexidade. Considere como exemplo duas strings $S1 = pqraxabcstvq$ e $S2 = xyaxbacsl$. Cada operação de Match (M) será pontuada com valor 2, cada Substituição (R) penalizada com valor -2, e cada operação de espaço (I ou D) penalizada com valor -1. Então, as substrings $\alpha = axabc$ e $\beta = axbac$ de $S1$ e $S2$ respectivamente possuem o alinhamento máximo exibido na tabela 4, com valor igual a 8.

Tabela 4 - Alinhamento Máximo

a	x	a	b	-	c	s
a	x	-	b	a	c	s

Ou seja, de todas as substrings possíveis existentes em $S1$ e $S2$, essas duas substrings são as que possuem o valor ótimo de similaridade (considerando esse esquema de pontuação para as operações M, R, I e D).

No presente trabalho optou-se pelo uso do Alinhamento Local ao invés da distância de edição (Alinhamento Global) por algumas questões observadas no problema. A principal delas está no fato de que quando se calcula a distância de edição de duas strings de tamanhos diferentes, muita penalização pode ocorrer em virtude das operações de Inserção ou Deleção, o que pode não ser desejado nesse problema.

Ilustrando um caso bastante pertinente dado o cenário observado. Imagine uma string $S1$ composta por 21 caracteres, e um algoritmo lhe diga que existe uma segunda string $S2$ onde o custo para se transformar $S1$ em $S2$ seja igual a 11 caracteres. Isto é, são necessários excluir de $S1$ 11 caracteres (mais da metade da string), para que $S1$ se torne $S2$. Isso não parece um bom cenário, mas é exatamente o que ocorrerá caso um usuário pergunte "Onde será a aula de Redes de Computadores?", e no Porteiro Web a disciplina apareça lançada como "Redes

Comp". Assim, em virtude de se ter muitas abreviações no Porteiro Web, escolher um algoritmo de Alinhamento Global (que penaliza strings de diferentes tamanhos), não seria uma escolha adequada ao problema.

Ao se analisar sob o aspecto do Alinhamento Local, no entanto, procurar por substrings que maximizem a similaridade é um caso adequado para quando se tem palavras inteiras (ditas pelo usuário), sendo comparadas com as mais diversas formas de abreviação do outro lado (dada a falta de padronização encontrada no Porteiro Web).

4.7.1.3 Adaptações ao algoritmo de Alinhamento Local e atribuição de pesos

Ainda sobre o problema abordado, um outro cenário interessante ocorre quando um usuário pergunta, por exemplo, "Onde será a aula de Fisiologia?". O problema é que no Porteiro Web a disciplina aparece registrada como "Fisio" (e isso dá um alinhamento local dos 5 primeiros caracteres). No entanto, ao se comparar com a disciplina de "Sociologia", o alinhamento será igual a 7 caracteres (mas dessa vez, em função da parte "iologia"). Assim, para minimizar esse tipo de problema, ajustou-se o algoritmo de alinhamento local com os seguintes pesos:

- Operação M
 - Caso ocorra no início de quaisquer das strings, pontua como +4;
 - Caso contrário, pontua como +2;
- Operação I, D, R
 - Pontua como -1

Dessa forma, "Fisiologia" resultaria em um alinhamento local igual a 10 pontos com "Fisio" (dadas as 5 operações M consecutivas feitas no início das strings); enquanto que resultaria em um alinhamento local igual a 7 pontos com "Sociologia" (pois temos 7 operações M ocorrendo no meio de ambas as strings). Como as abreviações geralmente são compostas pelo início das palavras, essa adequação ao algoritmo clássico (de pontuar mais no início da string) surtiu um efeito bastante positivo na qualidade dos resultados em observação empírica feita no laboratório.

4.8 Desenvolvimento do aplicativo

4.8.1 Protótipo

O primeiro passo foi a elaboração do protótipo da tela, com o objetivo de produzir um aplicativo bem intuitivo ao usuário e com ótima usabilidade. A pergunta pode ser feita tanto por entrada de voz, quanto por entrada de texto, preenchendo o campo destinado a ela, e podendo ser editada, posteriormente à fala ou à digitação.

A tela do aplicativo foi, então, projetada com objetivo de não ter sobrecarga de imagens, botões e afins, buscando ter um visual limpo, como mostra a Figura 16, onde tem-se o protótipo inicial do aplicativo, contendo apenas o botão de fala (microfone), o campo da pergunta, e o local destinado à exibição dos resultados, na parte inferior.

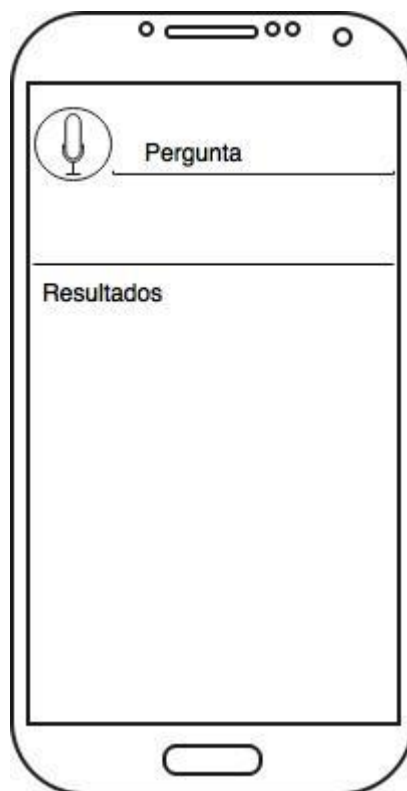


Figura 16 - Protótipo inicial da tela do aplicativo

A tela final teve o acréscimo de um botão que dispara a busca, necessário caso o usuário deseje digitar a pergunta ao invés de falar, ou para checar o

resultado da transformação de voz para texto, evitando buscas indesejadas, caso o reconhecedor de voz capture ou transforme alguma palavra em algo diferente do desejado pelo usuário.

4.8.2 Desenvolvimento Android

Os principais componentes de layout utilizados para a criação da tela foram:

RelativeLayout: Bastante versátil, permite posicionar um elemento em um local relativo a outro componente. Utilizado na tela principal do aplicativo.

LinearLayout: LinearLayout alinha todos os componentes nele contidos em uma única direção, vertical ou horizontal. Foi utilizado para a criação da exibição dos resultados em forma de lista.

Button: Um botão pode ser pressionado ou clicado pelo usuário para executar uma ação. Há dois botões neste aplicativo, em forma de ícones intuitivos, um para o usuário ativar o reconhecimento de voz (ícone de microfone) e o outro para disparar a busca (ícone de uma lupa).

EditText: É um campo de texto editável. Neste aplicativo, é destinado à pergunta do usuário.

ListView: Tem a capacidade de exibir itens em uma listagem. No caso deste projeto, o ListView exibe todos os resultados encontrados na busca.

ImageView: Apresenta uma imagem qualquer, tal como ícone e imagem. Há uma imagem de *background* com a logotipo do aplicativo, com o efeito “marca d’água”.

A Figura 17 demonstra, em destaque, alguns componentes presentes no aplicativo de forma geral.



Figura 17 - Tela do aplicativo, com destaque aos componentes de layout

Para realizar o reconhecimento de voz em Português Brasileiro, foi utilizado a API-Speech Recognize disponibilizada pelo próprio Android; e como já descrito, o usuário pode informar a pergunta através da fala ou simplesmente digitando.

Após a obtenção da pergunta, que no caso utilizaremos como exemplo a “Onde será a aula de Fisiologia?”, a URL destinada ao Web Service, para recuperar os resultados da busca, é montada como mostra a Figura 18:

104.255.220.109:8080/WebServiceEventosUnifal/?question=Onde%20será%20a%20aula%20de%20Fisiologia%3F

Figura 18 - Exemplo de URL para consulta ao Web Service

O Web Service então irá disponibilizar os resultados da busca utilizando JSON como formato de organização de dados.

Na Figura 19 é exibido o retorno dos resultados em JSON, ao consultar esta URL em um navegador comum, bem como seu resultado na tela do aplicativo:

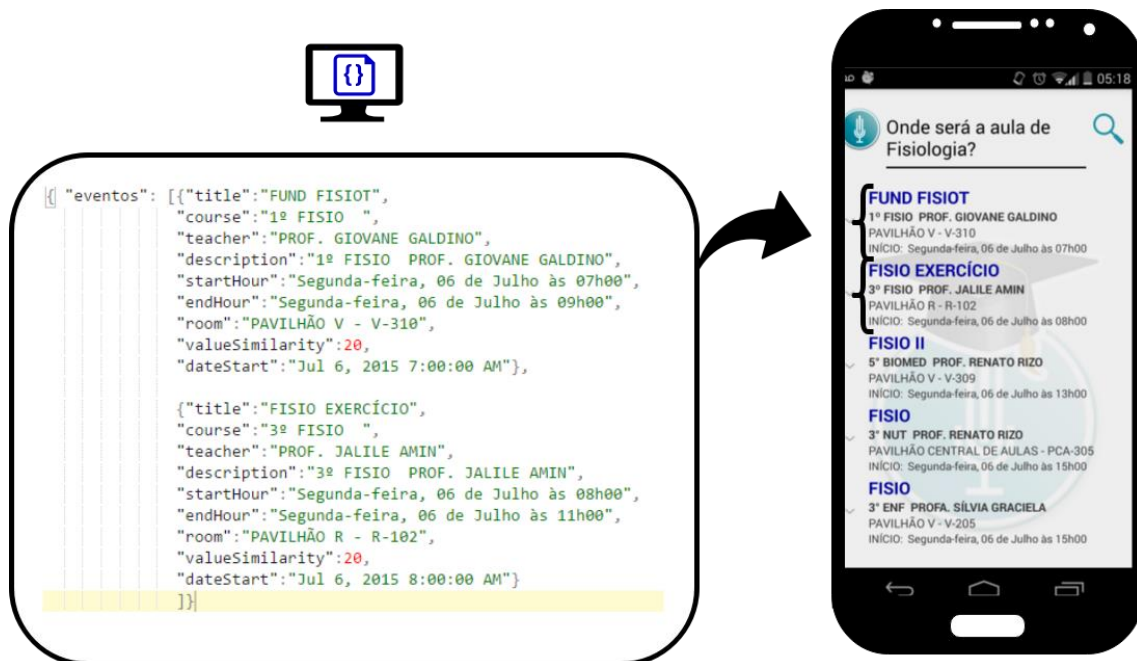


Figura 19 – Resultado do Web Service e da tela do aplicativo para uma pergunta específica

Os resultados recebidos do Web Service através do acesso via URL, são manipulados pelo aplicativo e exibidos no ListView da tela final do aplicativo, como também pode-se observar na Figura 19.

5

Resultados

Devido as estruturas utilizadas no processamento de identificação da similaridade, espera-se que o tempo de processamento da aplicação seja diretamente proporcional ao tamanho das strings a serem comparadas. Tendo isto em mente, foi feita uma breve análise em 3 meses de informações da base de dados extraída do sistema Porteiro Web, obtendo o tamanho mínimo, máximo e o mais recorrente dos títulos dos eventos.

Através destas informações, é possível estimar o tempo que a aplicação consome para trazer os resultados à tela, consultando como entrada para a aplicação, os menores, os maiores e os mais frequentes títulos de eventos existentes no banco.

Esta análise foi feita agrupando as strings do banco pelo tamanho dos títulos, quantizando a ocorrência de cada tamanho e somando os valores das ocorrências obtidas. Após isto, dividiu-se o valor desta soma, pelos valores das ocorrências, obtendo a probabilidade de ocorrência de strings com o respectivo tamanho.

Para obter o tempo de resposta, foram calculados a média aritmética dos resultados de 5 testes consecutivos realizados para cada tamanho de título diferente.

Nestes testes foram obtidos os seguintes valores: para títulos com 3 caracteres, obteve-se os tempos de respostas: 0,155; 0,177; 0,161; 0,163; 0,161; resultando em um tempo médio de 0,1634 segundos. Para títulos com 11 caracteres, obteve-se: 0,151; 0,159; 0,131; 0,198; 0,187; resultando em 0,1652 como tempo médio, e para títulos com 87 caracteres, obteve-se: 0,270; 0,257; 0,244; 0,237; 0,318; como tempos de respostas e 0,2526 como tempo médio de respostas. A tabela 5, demonstra esta análise.

Tabela 5 - Tamanho do título x Tempo de Resposta

Tamanho do Título	Quantidade de Ocorrência	Probabilidade de Ocorrência	Tempo de Resposta
3	34	0,230852797392721%	0,1634 segundos
11	2002	13,5931558935361%	0,1652 segundos
87	6	0,0407387289516567%	0,2526 segundos

Observando, a descrição acima e os dados da tabela 5, pode-se afirmar que o tempo de resposta está bastante satisfatório, mesmo para *strings* que sejam do tamanho máximo (87 caracteres) das encontradas no Banco de Dados.

6

Conclusão

Após a elaboração deste trabalho, alguns resultados podem ser considerados, levando em conta as problemáticas já conhecidas e encontradas ao longo da execução deste projeto.

O aplicativo desenvolvido apresentou-se ágil e eficaz, atendendo bem ao objetivo de redução do tempo de consulta de uma informação; enquanto que antes do desenvolvimento do mesmo, era necessário consultar o sistema, navegar entre as páginas do mesmo e realizar uma pesquisa, para só então obter a informação desejada, dispendendo tempo e trafegando uma grande quantidade de dados desnecessários. Além disto, também há o fato do sistema não ser responsivo, isto é, o Porteiro Web não foi desenvolvido visando melhorar a experiência de navegação em dispositivos móveis, piorando, assim, sua usabilidade.

Também nota-se a não popularidade do sistema entre os discentes, sendo o seu uso, mais frequente entre docentes e funcionários da instituição. Desta forma, alguns estudantes além de não acessar o sistema para obter as informações desejadas, deslocam-se até murais ou locais de fixação de informações, para ter acesso as mesmas.

Porém, com o advento deste aplicativo, espera-se amenizar estas dificuldades apresentadas, melhorando e agilizando o cotidiano das pessoas que frequentam e usufruem dos serviços da UNIFAL-MG.

7

Trabalhos Futuros

7.1 Restaurante Universitário (RU)

Inicialmente a proposta visava disponibilizar, além das informações do sistema Porteiro Web, informações a respeito do Restaurante Universitário (RU), como cardápio e horários de funcionamento, porém, atualmente seu HTML não apresenta boa estrutura e padronização, impossibilitando a extração das informações do mesmo através do Web Crawler. Desta forma, tem-se como parte do trabalho futuro a inclusão das informações do Restaurante Universitário na aplicação desenvolvida neste trabalho.

7.2 Padronização do sistema “Porteiro Web”

O modo como as informações são inseridas no sistema Porteiro Web, truncadas, abreviadas e até com erros de digitação, atende, mesmo que parcialmente, o objetivo do sistema, que é gerenciar informações acerca de eventos, nos espaços da universidade; pois até mesmo com a facilidade do ser humano em interpretar siglas e truncamentos, pode haver dificuldade, por parte de algumas pessoas menos instruídas ou até menos alfabetizadas.

Além disto, a falta de padronização do sistema implicou em várias dificuldades enfrentadas durante este trabalho, que foram amenizadas computacionalmente através de um algoritmo de similaridade; porém, mesmo com este artifício, há informações que estão truncadas de tal forma que impossibilitaram a capacidade do algoritmo de identificar similaridade, caracterizando perda de informação e tornando a qualidade do aplicativo duvidosa.

Para sanar todos estes problemas, pretende-se comunicar a necessidade de padronização das informações do porteiro web ao núcleo de tecnologia da informação da universidade, responsável pela administração do sistema. Desta forma, espera-se que futuramente, o aplicativo tenha a desejada abrangência de busca plena por todas as informações contidas no Porteiro Web.

7.3 Expansão do aplicativo

Futuramente, pretende-se expandir as funcionalidades do aplicativo para todos os outros Campus da UNIFAL-MG que ainda não foram abrangidos na presente versão. Os Campus restantes são os Campus de Poços de Caldas e de Varginha.

Como já citado, houve o desejo de integrar as informações do Restaurante Universitário dos Campus de Alfenas (Sede) ao aplicativo. Porém, ainda há uma grande pretensão futura a este respeito, integrando, além destas, as informações pertinentes ao Restaurante Universitário do Campus de Poços de Caldas, abrangendo, desta forma todos os restaurantes existentes na instituição.

8

Referências Bibliográficas

AHO, Alfred V.; SETHI, Ravi; ULLMAN, Jeffrey D. **Compilers, Principles, Techniques**. Addison wesley, 1986.

ALTSCHUL, Stephen F. et al. Basic local alignment search tool. **Journal of molecular biology**, v. 215, n. 3, p. 403-410, 1990.

BLOOMBERG BUSINESS ,<<http://www.bloomberg.com/bw/stories/2005-08-16/google-buys-android-for-its-mobile-arsenal>>. Acesso em: 10/06/2015 às 16h.

DE PAULA, Leonam João Leal. **Desenvolvimento de aplicativo para dispositivos móveis para coleta de dados georreferenciados através de reconhecimento de voz**. 2013. PhD Thesis. Escola Superior de Agricultura “Luiz de Queiroz.

DEVELOPERS, **Android**. What is android. 2011.

FERNANDES, Marcio Merino; PIZZOLATO, Ednaldo Brigante. **Interface de comando de voz para sistemas de Realidade Aumentada**. In: Anais do I Workshop de Realidade Aumentada. 2004. p. 21-24.

GOOGLE, <<https://www.google.com/webmasters/tools/mobile-friendly/>>. Acesso em 07/06/2015 às 20h.

GUSFIELD, Dan. **Algorithms on strings, trees and sequences: computer science and computational biology**. Cambridge university press, 1997.

HAMAD, Hatem; SAAD, Motaz; ABED, Ramzi. **Performance Evaluation of RESTful Web Services for Mobile Devices**. *Int. Arab J. e-Technol.*, 2010, 1.3: 72-78.

ISHIBASHI, Larissa Tiemi Fujii; RODRIGUES, Rayanne Viegas; DE OLIVEIRA, Elton Tamiozzo. **Aplicativo para Dispositivos Móveis do Sistema Integrado de Informações Acadêmicas da Universidade Católica Dom Bosco.**

JADHAV, A.; PATIL, Arvind. Android Speech To Text Converter for SMS Application. **IOSR Journal of Engineering**, v. 2, n. 3, p. 420-423, 2012.

JSON, <<http://www.json.org>> . Acesso em: 14/06/2015 às 9h

KOBERSTEIN, Jonathan; NG, Yiu-Kai. **Using word clusters to detect similar web documents.** In: *Knowledge Science, Engineering and Management*. Springer Berlin Heidelberg, 2006. p. 215-228.

MACIEL, Alexandre, et al. **Desenvolvendo soluções com interface baseada em voz.** In: Companion Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems. Brazilian Computer Society, 2012. p. 59-62.

NURSEITOV, Nurzhan, et al. **Comparison of JSON and XML Data Interchange Formats: A Case Study.** *Caine*, 2009, 9: 157-162.

PANTA, Deepak, et al. **Web crawling and scraping: developing a sale-based website.** 2015.

PINKERTON, Brian. **Finding what people want: Experiences with the WebCrawler.** In: Proceedings of the Second International World Wide Web Conference. 1994. p. 17-20.

SANNER, Michel F., et al. **Python: a programming language for software integration and development.** *J Mol Graph Model*, 1999, 17.1: 57-61.

SNOWBALL, <<http://snowball.tartarus.org/algorithms/portuguese/stop.txt>>. Acesso em: 20/06/2015 às 12h.

SRIVASTAVA, Biplav; KOEHLER, Jana. **Web service composition-current solutions and open problems.** In: *ICAPS 2003 workshop on Planning for Web Services*. 2003. p. 28-35.

TECMUNDO, <<http://www.tecmundo.com.br/google-i-o-2015/80678-reconhecimento-voz-google-tem-so-8-erro-nao-de-melhorar.htm>>. Acesso em 10/06/2015 às 13h.

TESTE DE COMPATIBILIDADE COM DISPOSITIVOS MÓVEIS, <https://www.google.com/webmasters/tools/mobile-friendly/>. Acesso em 20/06/2015 às 14h.

VERTANEN, Keith; KRISTENSSON, Per Ola. **Parakeet: A continuous speech recognition system for mobile touch-screen devices**. In: *Proceedings of the 14th international conference on Intelligent user interfaces*. ACM, 2009. p. 237-246.

WALKER, Willie, et al. **Sphinx-4: A flexible open source framework for speech recognition**. 2004.

WANG, Jing; GUO, Yuchun. **Scrapy-Based Crawling and User-Behavior Characteristics Analysis on Taobao**. In: *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2012 International Conference on*. IEEE, 2012. p. 28-35.

WIKIPEDIA, <<http://pt.wikipedia.org/wiki/URI>>. Acesso em: 11/06/2015 às 18h.

WIKIPEDIA, <http://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_din%C3%A2mica>. Acesso em: 11/06/2015 às 13h.